Published in Journals: Geomatics, ISPRS International Journal of Geo-Information and Remote Sensing

Topic Reprint

Geocomputation and Artificial Intelligence for Mapping

Edited by
Lili Jiang, Di Zhu and An Zhang

mdpi.com/topics

MDPI

Geocomputation and Artificial Intelligence for Mapping

Geocomputation and Artificial Intelligence for Mapping

Topic Editors

Lili Jiang
Di Zhu
An Zhang

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Topic Editors

Lili Jiang
State Key Laboratory of Resources and Environmental Information System
Chinese Academy of Sciences
Beijing
China

Di Zhu
Department of Geography, Environment and Society
University of Minnesota
Twin Cities, MN
USA

An Zhang
State Key Laboratory of Resources and Environmental Information System
Chinese Academy of Sciences
Beijing
China

Editorial Office

MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Topic, published open access by the journals *Geomatics* (ISSN 2673-7418), *ISPRS International Journal of Geo-Information* (ISSN 2220-9964) and *Remote Sensing* (ISSN 2072-4292), freely accessible at: https://www.mdpi.com/multidisciplinary_topics/geo_computation.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-9183-2 (Hbk)
ISBN 978-3-7258-9184-9 (PDF)
https://doi.org/10.3390/books978-3-7258-9184-9

© 2026 by the authors. Articles in this reprint are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The reprint as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (https://creativecommons.org/licenses/by-nc-nd/4.0/).

Contents

Rui Deng, Yanning Guan, Danlu Cai, Tao Yang, Klaus Fraedrich, Chunyan Zhang, et al.
Supervised versus Semi-Supervised Urban Functional Area Prediction: Uncertainty, Robustness and Sensitivity
Reprinted from: *Remote Sens.* **2023**, *15*, 341, https://doi.org/10.3390/rs15020341 . . . . . **1**

Jia Song and Xiangbing Yan
The Effect of Negative Samples on the Accuracy of Water Body Extraction Using Deep Learning Networks
Reprinted from: *Remote Sens.* **2023**, *15*, 514, https://doi.org/10.3390/rs15020514 . . . . . **24**

Ding Yu, Aihua Li, Yinping Long, Yan Xu, Jinrui Li and Xiongwu Xiao
Mean Inflection Point Distance: Artificial Intelligence Mapping Accuracy Evaluation Index—An Experimental Case Study of Building Extraction
Reprinted from: *Remote Sens.* **2023**, *15*, 1848, https://doi.org/10.3390/rs15071848 . . . . . **41**

Ke Zhou, Zhengyan Zhang, Le Liu, Ru Miao, Yang Yang, Tongcan Ren and Ming Yue
Research on SUnet Winter Wheat Identification Method Based on GF-2
Reprinted from: *Remote Sens.* **2023**, *15*, 3094, https://doi.org/10.3390/rs15123094 . . . . . **63**

Yaxin Chen, Xin Shen, Guo Zhang and Zezhong Lu
Multi-Objective Multi-Satellite Imaging Mission Planning Algorithm for Regional Mapping Based on Deep Reinforcement Learning
Reprinted from: *Remote Sens.* **2023**, *15*, 3932, https://doi.org/10.3390/rs15163932 . . . . . **81**

Qingyong Zhang, Lingfeng Zhou, Yixin Su, Huiwen Xia and Bingrong Xu
Gated Recurrent Unit Embedded with Dual Spatial Convolution for Long-Term Traffic Flow Prediction
Reprinted from: *ISPRS Int. J. Geo-Inf* **2023**, *12*, 366, https://doi.org/10.3390/ijgi12090366 . . . . **103**

Yihao Sun, Han Hu, Yawen Han, Ziyan Wang and Xiaodi Zheng
Large-Scale Automatic Identification of Industrial Vacant Land
Reprinted from: *ISPRS Int. J. Geo-Inf* **2023**, *12*, 409, https://doi.org/10.3390/ijgi12100409 . . . . **125**

Jingxue Wang, Huan Li, Zhenghui Xu and Xiao Xie
Semantic Segmentation of Urban Airborne LiDAR Point Clouds Based on Fusion Attention Mechanism and Multi-Scale Features
Reprinted from: *Remote Sens.* **2023**, *15*, 5428, https://doi.org/10.3390/rs15215248 . . . . . **142**

Ali Polat, İnan Keskin and Özlem Polat
Automatic Detection and Mapping of Dolines Using U-Net Model from Orthophoto Images
Reprinted from: *ISPRS Int. J. Geo-Inf* **2023**, *12*, 456, https://doi.org/10.3390/ijgi12110456 . . . . **166**

Mingyang Yu, Haiqing Xu, Fangliang Zhou, Shuai Xu and Hongling Yin
A Deep-Learning-Based Multimodal Data Fusion Framework for Urban Region Function Recognition
Reprinted from: *ISPRS Int. J. Geo-Inf* **2023**, *12*, 468, https://doi.org/10.3390/ijgi12120468 . . . . **182**

Mohamed A. Damos, Jun Zhu, Weilian Li, Elhadi Khalifa, Abubakr Hassan, Rashad Elhabob, et al.
Enhancing the K-Means Algorithm through a Genetic Algorithm Based on Survey and Social Media Tourism Objectives for Tourism Path Recommendations
Reprinted from: *ISPRS Int. J. Geo-Inf* **2024**, *13*, 40, https://doi.org/10.3390/ijgi13020040 . . . . . **203**

Alireza Hajiheidari, Mahmoud Reza Delavar and Abbas Rajabifard
Smart Urban Cadastral Map Enrichment—A Machine Learning Method
Reprinted from: *ISPRS Int. J. Geo-Inf* **2024**, *13*, 80, https://doi.org/10.3390/ijgi13030080 . . . . . **221**

Yan Fu, Qingwen Qi, Lili Jiang and Yapeng Zhao
Spatial Process Analysis of the Evolution of Farmland Landscape in China
Reprinted from: *ISPRS Int. J. Geo-Inf* **2024**, *13*, 98, https://doi.org/10.3390/ijgi13030098 . . . . . **246**

Joseph K. Geisz, Phillipe A. Wernette and Peter C. Esselman
Classification of Lakebed Geologic Substrate in Autonomously Collected Benthic Imagery Using Machine Learning
Reprinted from: *Remote Sens.* **2024**, *16*, 1264, https://doi.org/10.3390/rs16071264 . . . . . **261**

Thomas H. Meyer
Vector-Algebra Algorithms to Draw the Curve of Alignment, the Great Ellipse, the Normal Section, and the Loxodrome
Reprinted from: *Geomatics* **2024**, *4*, 8, https://doi.org/10.3390/geomatics4020008 . . . . . **290**

Zehua An, Xiaoping Rui and Chaojie Gao
Improved A* Navigation Path-Planning Algorithm Based on Hexagonal Grid
Reprinted from: *ISPRS Int. J. Geo-Inf* **2024**, *13*, 166, https://doi.org/10.3390/ijgi13050166 . . . . **301**

Kaixuan Du, Fu Ren, Yong Wang, Xianghong Che, Jiping Liu, Jiaxin Hou and Zewei You
Integration of Spatial and Co-Existence Relationships to Improve Administrative Region Target Detection in Map Images
Reprinted from: *ISPRS Int. J. Geo-Inf* **2024**, *13*, 216, https://doi.org/10.3390/ijgi13060216 . . . . **315**

Yaqi Huang, Minrui Zheng, Tianle Li, Fei Xiao and Xinqi Zheng
An Integrated Framework for Landscape Indices' Calculation with Raster–Vector Integration and Its Application Based on QGIS
Reprinted from: *ISPRS Int. J. Geo-Inf* **2024**, *13*, 242, https://doi.org/10.3390/ijgi13070242 . . . . **333**

Yu Wang, Shunping Zhou, Qingfeng Guan, Fang Fang, Ni Yang, Kanglin Li and Yuanyuan Liu
Enhancing Place Emotion Analysis with Multi-View Emotion Recognition from Geo-Tagged Photos: A Global Tourist Attraction Perspective
Reprinted from: *ISPRS Int. J. Geo-Inf* **2024**, *13*, 256, https://doi.org/10.3390/ijgi13070256 . . . . **352**

Kai Wu, Zugang Chen, Xinqian Wu, Guoqing Li, Jing Li, Shaohua Wang, et al.
Extracting Geoscientific Dataset Names from the Literature Based on the Hierarchical Temporal Memory Model
Reprinted from: *ISPRS Int. J. Geo-Inf* **2024**, *13*, 260, https://doi.org/10.3390/ijgi13070260 . . . . **375**

Zhiwei Wu, Tian Lan, Chenzhen Sun, Donglin Cheng, Xing Shi, Meisheng Chen and Guangjun Zeng
An Improved ANN-Based Label Placement Method Considering Surrounding Features for Schematic Metro Maps
Reprinted from: *ISPRS Int. J. Geo-Inf* **2024**, *13*, 294, https://doi.org/10.3390/ijgi13080294 . . . . **398**

Haohua Zheng, Jianchen Zhang, Heying Li, Guangxia Wang, Jianzhong Guo and Jiayao Wang
Road Network Intelligent Selection Method Based on Heterogeneous Graph Attention Neural Network
Reprinted from: *ISPRS Int. J. Geo-Inf* **2024**, *13*, 300, https://doi.org/10.3390/ijgi13090300 . . . . **419**

Grayson R. Morgan, Danny Zlotnick, Luke North, Cade Smith and Lane Stevenson
Deep Learning for Urban Tree Canopy Coverage Analysis: A Comparison and Case Study
Reprinted from: *Geomatics* **2024**, *4*, 22, https://doi.org/10.3390/geomatics4040022 . . . . . **445**

Kauê de Moraes Vestena, Silvana Phillipi Camboim, Maria Antonia Brovelli and Daniel Rodrigues dos Santos
Investigating the Performance of Open-Vocabulary Classification Algorithms for Pathway and Surface Material Detection in Urban Environments
Reprinted from: *ISPRS Int. J. Geo-Inf* **2024**, *13*, 422, https://doi.org/10.3390/ijgi13120422 . . . . **466**

remote sensing

MDPI

Article

Supervised versus Semi-Supervised Urban Functional Area Prediction: Uncertainty, Robustness and Sensitivity

Rui Deng [1,2], Yanning Guan [1,*], Danlu Cai [1], Tao Yang [3], Klaus Fraedrich [4], Chunyan Zhang [1], Jiakui Tang [5,6], Zhouwei Liao [7], Zhishou Wei [1,2] and Shan Guo [1]

1 Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100101, China
2 University of Chinese Academy of Sciences, Beijing 100049, China
3 The School of Architecture, Tsinghua University, Beijing 100084, China
4 Max Planck Institute for Meteorology, 20146 Hamburg, Germany
5 College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China
6 Yanshan Earth Key Zone and Surface Flux Observation and Research Station, University of Chinese Academy of Sciences, Beijing 101408, China
7 Yangtze River Basin Operation Management Center, China Three Gorges, Co., Ltd., Yichang 443133, China
* Correspondence: guanyn@radi.ac.cn

Abstract: To characterize a community-scale urban functional area using geo-tagged data and available land-use information, several supervised and semi-supervised models are presented and evaluated in Hong Kong for comparing their uncertainty, robustness and sensitivity. The following results are noted: (i) As the training set size grows, models' accuracies are improved, particularly for multi-layer perceptron (MLP) or random forest (RF). The graph convolutional network (GCN) (MLP or RF) model reveals top accuracy when the proportion of training samples is less (greater) than 10% of the total number of functional areas; (ii) With a large amount of training samples, MLP shows the highest prediction accuracy and good performances in cross-validation, but less stability on same training sets; (iii) With a small amount of training samples, GCN provides viable results, by incorporating the auxiliary information provided by the proposed semantic linkages, which is meaningful in real-world predictions; (iv) When the training samples are less than 10%, one should be cautious using MLP to test the optimal epoch for obtaining the best accuracy, due to its model overfitting problem. The above insights could support efficient and scalable urban functional area mapping, even with insufficient land-use information (e.g., covering only ~20% of Beijing in the case study).

Keywords: geotagged data; urban functional area; graph convolutional network; multi-layer perceptron; semi-supervised learning

1. Introduction

Land use is defined by the function, or functions, characterizing humans' use of an area of land, which mostly falls within six main categories, including agricultural, residential, recreational, commercial, industrial and transportation [1], in which the last four are the main contributors for urban area coverage. Commercial land use is land being used for the sale of goods or services for financial profit, including central business districts and shopping centers. Residential land use is the land used for housing. Recreational land use in urban areas includes city parks, playing fields, hiking and biking trails, etc. Industrial land use is the land used in manufacturing and storing, etcetera. Transport land use is the land delegated to the moving or transportation of goods and commuting people from one spot to another; that is, roads, highways, railroads and airports.

Urbanization is a process whereby populations move from rural to urban areas, enabling cities and towns to grow [2,3], and the process typically brings the need for more housing and jobs, associating with a need for land use change. That is, agricultural or natural recreational types of land use must be converted to residential, business, industrial

and transportation types [4,5]. During the renewal of a city, different functional areas with relatively homogeneous internal functional land use are gradually formed to meet the various needs of people's daily life and they are considered as basic spatial units implementing urban plans [6]. The combination and distribution of different areas constitute the structure of the city [7].

Efforts have been made to monitor real urban functional land use patterns [8], and to compare with the urban development plan, which is important for sustainable urban development. Some of the urban functional areas may be inevitably sculpted by anthropogenic activities, rather than following the initial masterplan, due to multiple driving factors, including local geographic/topographic conditions [9,10], economic purposes [11,12], etc. Thus, diagnosing differences between plan and reality enables an improved future urban planning, such as modifications for planning transportation and recreational spaces [13,14]. Besides, detailed urban functional maps and the dynamics of urban functional area change make a possible estimation for urban resilience in facing natural hazards or climate change [14–18], and for the ecological impact, due to urban sprawl, developing pattern, clustering, trend and functional land use interrelationships [19–24].

Nevertheless, before commencing all those analyses, a precise knowledge of the distribution of urban functional areas is necessary, which depends on the classification model, the amount of training data, different combinations of training data, etc. The unsupervised clustering methods exhibit the ability to discover various functional areas, yet ultimately require manual identification of the properties of individual clusters, which could bring about large uncertainty [25]. Recently, artificial intelligence (AI) and machine learning (ML) developers have generated AI and ML to "think more intelligently", like humans, making decisions with supervised and semi-supervised models on urban functional area analysis [26–28]. Later, GeoAI and artificial intelligence, together with a geographical information system (GIS), performs well in numerous tasks via combining the strong modeling ability of AI and geospatial characteristics (see for example [29–32]). Still, less effort has been made for diagnosing the sensitivities of performances which are related to the selection of the classification models, discrepancies in training data sets, and the method for quantifying model uncertainty. Previous works assess classification accuracies by comparing results with planning maps and online maps (e.g., [28,33–36]), or using land use attributes extracted from human activities [27], but very few have leveraged standardized land use datasets with non-empirical means to determine the viability of classification models [37].

Research using geotagged training data adds new perspectives to data mining and to categorizing urban functional land use on a community scale. Geotagging is the process of appending geographic coordinates to media-based on the location of a mobile device [38]. Geotags usually consist of coordinates, such as latitude and longitude, but may also include bearing, altitude and place names, which can be applied to photos, videos, or QR codes, and could also include time stamps or other contextual information [38]. The point of interest (POI) data is one of the most popular geo-tagged data [26,39], used for sensing urban functional area characteristics associated with analysis methods based on GIS and GeoAI [8,37,40].

However, data requirements in number and quality are different for diverse supervised and semi-supervised ML models [41]. For example, traditional non-graph structured supervised models may not capture the complex interactions and connections between urban functional elements [30], which results in a larger requirement of training data. In comparison, graph-based classification models require relatively less training samples (i.e., support a semi-supervised learning) due to their ability to combine semantic linkages between urban functional areas [27,30]. Efficient methods, however, for building linkages are essential for the performance of graph-based classification models [27].

In this study, first, the basic study unit (i.e., functional area) is designed in Hong Kong, according to urban road networks, to obtain a relatively more homogeneous functional property (see [36,42]), followed by data-labeling on the basic study unit, which is the process

of identifying raw data and adding meaningful labels to provide context for machine learning. After applying the supervised and semi-supervised ML models to achieve the urban functional area classification, the models' performances, in terms of uncertainty, robustness and sensitivity, will be compared to give insights into model selection strategies for different scenarios. The resulted insights are further supported by a case study carried out in Beijing.

2. Materials and Methods

Three data sets are used as inputs: the point of interest (POI) dataset, the open street map (OSM) dataset and a public land use dataset. The POI data used in this study is collected from the Amap platform, which consists of attributes including longitudes, latitudes, type (see Appendix A, Table A1) and rating scores, etc., which provide information about real-world geographical locations on which human activities take place [10]. A Python language-based web crawler is developed for accessing and storing the JSON-formatted POI data: (https://gitee.com/pickup20/multi-modal-paper/tree/mastere/data, POI data was accessed on 1 March 2022). The OSM dataset includes detailed world-wide road network and part of the land use information. The road network consists of multiple classes of roads, of which the classes of primary, secondary and tertiary are used. The main drawbacks of land use information in OSM data are the inadequacy in some areas and incorrect records. The recently updated OSM data is downloaded from https://www.openstreetmap.org/ accessed on 3 April 2022. Finally, the land use dataset of Hong Kong (Land Utilization in Hong Kong, LUHK) with 10 × 10 m spatial resolution is used for testing the classification accuracy, which is downloaded from https://www.pland.gov.hk/pland_en/info_serv/open_data/landu/ accessed on 16 April 2022.

The identification and classification scheme of an urban functional area (see Figure 1) consists of commercial, residential, public service, recreational and transportation. Here, for diagnosing more details about urban central areas, agricultural and industrial urban functional land use types mentioned in the first chapter are excluded and replaced by public service land use type, including schools, institutions and administration facilities.

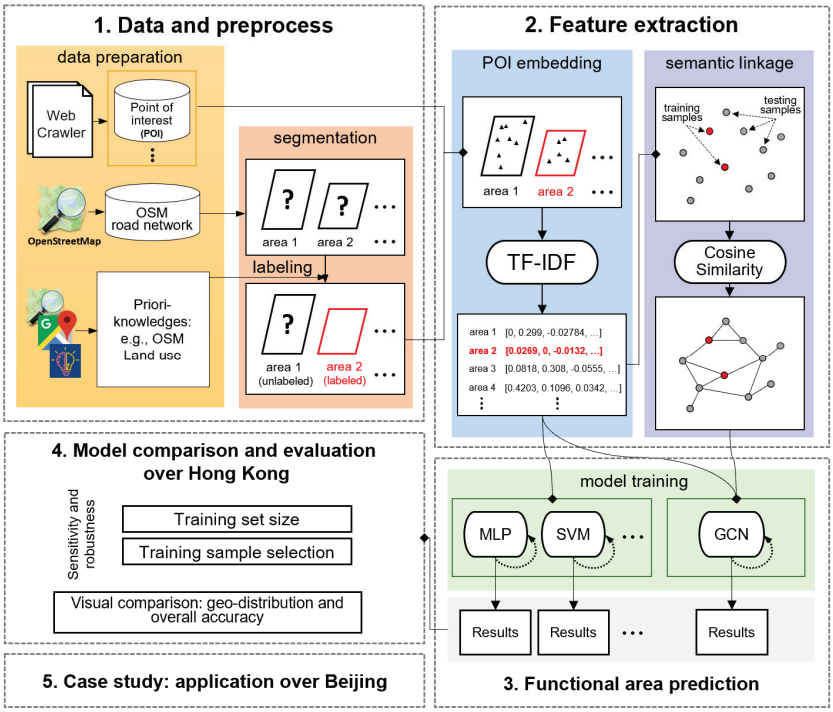


Figure 1. A flowchart illustrating the five stages of the functional area type identification and classification conducted in this paper.

2.1. Preprocessing

The input data (e.g., OSM and POI) attain a unified coordinate system by map projection transformation as the known functional land use datasets; then, they are reclassified into a unified classification scheme (see Table 1) for further categorizing, feature extraction, semantic linkage and model comparison. The following processes are introduced:

Segmentation: Urban road network data is used to divide the study area into basic study units (i.e., areas), according to the segmentation method proposed by [42]. The initial OSM road networks are line vector features. Followed by transforming the line vectors to polygon vectors (using the ArcGIS 10.7 software), polygons with an area less than 0.001 km^2 are merged into the closest polygon through a morphological closing manipulation, which eliminates the scattered objects while preserving the shapes and sizes of larger objects.

Labeling: The segmented polygons are labeled/categorized (using the Zonal Statistic Tool in ArcGIS 10.7), by retrieving the functional land use type with the highest coverage within them. Occupation percentage of the labeled class is calculated for further possible bias testing (see Section 3.1.4). Note that in our case study, only when the highest coverage of the functional land use type exceeds 50%, a valid functional area is attained; otherwise, the function of this area remains unknown.

Table 1. Functional land use types in this study associated with OSM functional land use dataset and the Land Utilization in Hong Kong (LUHK) dataset.

Functional Land Use Types in This Paper	OSM Land Use Category	Land Utilization in Hong Kong (LUHK) Category
Commercial	Retail	Commercial/Business and Office
Residential	Residential	Private Residential Public Residential Rural Settlement
Public service	University Museum Public	Government, Institutional and Community Facilities
Recreational	Garden Leisure Park Recreation Ground	Open Space and Recreation
Transportation	Railway	Roads and Transport Facilities Railways Airport Port Facilities
Not available (NA)	Other	Other

2.2. Feature Extraction

2.2.1. POI Embedding for Functional Area Vectorizing

Representation methods, such as the Word2Vec and the Term Frequency-Inverse Document Frequency (TF-IDF), in the natural language-processing field, enable the parse or process of natural language to a standard feature vector format for present models. Here, the Term Frequency-Inverse Document Frequency (TF-IDF) is used to obtain a standard feature vector for each functional area from its related POI categories.

Importance of POI categories within the functional area (represented by the TF value): The frequency of occurrence for the *i*th POI category, POI_i, providing a weight of its importance within the specific functional area, which is calculated as:

$$TF_{AREA_j,\, POI_i} = \frac{Freq(AREA_j,\, POI_i)}{\sum_{i=0}^{C} Freq(AREA_j,\, POI_i)} \tag{1}$$

where *AREA* represents one specific functional area out of all areas, with a total number of *N*, and *j* is a counting integer ranging from 0 to *N*. $Freq(AREA_j,\, POI_i)$ is the number of occurrences of a specific POI category (POI_i) in a given area ($AREA_j$), with the integer *i* ranging from 0 to *C*.

Importance of POI category among all functional areas (represented by the IDF value): the Inverse Document Frequency (IDF) value of a given POI category, IDF_{POI_i}, is calculated as:

$$IDF_{POI_i} = ln\frac{N}{1+\sum_{j=0}^{N} IsPresent\left(AREA_j,\ POI_i\right)} \tag{2}$$

where:

$$IsPresent\left(AREA_j,\ POI_i\right) = \begin{cases} 1,\ Freq\left(AREA_j,\ POI_i\right) > 0 \\ 0, Freq\left(AREA_j,\ POI_i\right) = 0 \end{cases} \tag{3}$$

This value indicates the overall view of a POI category in the whole study region. That is, the more frequently the POI category occurs among all the functional areas, the lower is the IDF value.

Combination (TF-IDF): Thus, a normalized Term Frequency-Inverse Document Frequency (TF-IDF) feature vector for each functional area, $TFIDF_{AREA_j}$, can be calculated for a further supervised or semi-supervised classification:

$$TFIDF_{AREA_j} = \frac{1}{\sqrt{\sum_{i=0}^{C}\left(TF_{AREA_j,\ POI_i}\cdot IDF_{POI_i}\right)^2}}\left(TF_{AREA_j,\ POI_0}\cdot IDF_{POI_0},\ TF_{AREA_j,\ POI_1}\cdot IDF_{POI_1}, \ldots, TF_{AREA_j,\ POI_C}\cdot IDF_{POI_C}\right)^T \tag{4}$$

The similarity among TF-IDF feature vectors from each functional area is also used for a further mapping of the land use distribution (detailed information see Section 2.2.2).

2.2.2. Similarity Measurement for Functional Area Semantic Linkage

Pairwise relations or linkages of functional areas have been considered to be useful information in big data based urban computation [27]. Together with the TF-IDF feature vector of the functional areas, a graph could be obtained for further exploration of the implicit information among all functional areas, so as to improve the final urban functional area classification. The linkages could be simply derived by the normalized TF-IDF feature vectors.

Semantic similarity: A graph consists of nodes and their linkages. Here, the nodes are areas with the associated normalized TF-IDF feature vectors $TFIDF_{AREA}$, and the linkages are calculated as the cosine similarity of TF-IDF feature vectors from two different areas, $Sim(TFIDF_{AREA_m}, TFIDF_{AREA_n})$:

$$Sim(TFIDF_{AREA_m}, TFIDF_{AREA_n}) = \frac{TFIDF_{AREA_m}\cdot TFIDF_{AREA_n}}{\|TFIDF_{AREA_m}\|\cdot\|TFIDF_{AREA_n}\|} \tag{5}$$

where m, n representing different functional areas, are integers ranging from 0 to N. Threshold is set as 0.7 for separating the final interdependencies or similarities among all the functional areas (see similar link prediction strategy in [43]). Values exceeding the threshold indicate that there is a noticeable interrelationship between two functional areas. The final linkages are organized by an adjacency matrix, A, with each element calculated as:

$$A_{m,n} = \begin{cases} Sim(AREA_m, AREA_n),\ Sim(AREA_m, AREA_n) > 0.7 \\ 0,\ Sim(AREA_m, AREA_n) \leq 0.7 \end{cases} \tag{6}$$

2.3. Urban Functional Area Prediction Models

Both supervised and semi-supervised models are commonly used for classification tasks. The differences here are that only the normalized TF-IDF feature vectors (calculated from Section 2.2.1) are required by supervised models, e.g., support vector machine (SVM, [44]), random forest (RF, [45]) and multi-layer Pperceptron (MLP, [46]), while both the normalized TF-IDF feature vectors and their linkages (Section 2.2.2) should serve as input for semi-supervised models, e.g., graph convolutional network (GCN, [41]), and comparison (see Table 2). Followed by an introduction, five models (four supervised versus one semi-supervised) are built and compared in the application analysis (Section 3).

Table 2. Characteristics and implementation details of 5 models compared in this study.

Model		Training Sample Quantity	Hyperparameters	Python Library
Supervised	SVM	Large	Value of the slack variable (In Soft Margin term)	Scikit-learn
	RF-128 RF-200	Large	Number of Decision Trees	Scikit-learn
	MLP	Large	(a) Number of Hidden Layers (b) Corresponding Neurons (c) Weighting and Focusing Factors (In Focal Loss Function)	Pytorch
Semi-supervised	GCN	Relatively Less	Similarity Thresholds (In Semantic Linkage)	Pytorch-Geometric

2.3.1. Supervised Models

Support Vector Machine (SVM): A SVM is a supervised machine learning algorithm used for both classification and regression, which has become exceedingly popular, due to its relative simplicity and flexibility in addressing a range of classification problems, even in studies where sample sizes may be limited [44,46]. After providing the SVM model sets of labeled training data for each category, it defines a decision boundary (i.e., a hyperplane) by maximizing the width of the gap between two categories to best separate them. Note that SVM is a binary classification method, which classifies objects into two groups of "True" and "False". Thus, a simple workaround of One-vs.-Rest is implemented to obtain a multi-class classification (for details, see [47]).

Random Forest (RF): RF is a classic ensemble learning model, which is built based on decision trees to provide results about modeling predictions and behavior analysis [45]. Each decision tree in RF represents a distinct instance of the classification of data input into the random forest by integrating the entropy function to measure the loss between the prediction and the true label. RF considers the instances individually, taking the one with the majority of votes from decision trees as the selected prediction.

Multi-layer Perceptron (MLP): MLP is a widely applied supervised neural network model, which consists of three types of layers: (a) the input layer receives the input signal to be processed; (b) the output layer displays results from the required task, e.g., prediction and classification; (c) the hidden layers (with arbitrary number) are placed in between the input and output layer, which are the true computational engine of the MLP [46]. Training data flows in the forward direction from input to output layer, and the neurons in the MLP are trained with the back propagation learning algorithm; thus, they approximate any continuous function for unknown pattern classification, recognition, prediction and approximation. Note that a focal loss function (proposed by Lin, et al. [48]) is used to address the class imbalance problem during the training process [37,49], which poses a problem in machine learning when the numbers of training samples for different classes vary greatly:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma} log(p_t) \tag{7}$$

where t represents the classified type. The focal loss adds a factor, $\alpha_t(1-p_t)^{\gamma}$, to the standard cross entropy criterion, $-log(p_t)$, to reduce the relative loss for well-classified examples. Here, p_t is the softmax-normalized t-th output of the model, and α_t is a weighting factor corresponding to the model's t-th output. γ is for reducing well-classified examples loss, thus, forcing the model to focus on hard and misclassified objects, thereby improving the model performance.

2.3.2. Semi-Supervised Model: Graph Convolutional Network

Unlike traditional machine learning (e.g., SVM, RF, MLP) lacking consideration of graph structured semantic linkage, graph-based models have the potential to predict urban land use types with a small number of training data, which is meaningful in the real-world

prediction. This could be achieved via measuring the semantic linkages or their similarity distances, etc. [27]. The requirements of training data are largely reduced for these models compared to others, i.e., they could be trained in a semi-supervised manner.

Kipf and Welling [41] proposed a multi-layer graph convolutional network to scale linearly in the number of graph edges and learn hidden layer representations that encode both features of nodes and the graph structure. For a GCN model with L hidden layers, the forward propagation rule of graph convolution is given by:

$$H^{(l+1)} = f\left(H^{(l)}, A\right) = \sigma\left(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{8}$$

where $H^{(0)}$ represents *TF-IDF* in each area, and $H^{(l)}$ is for i-th neural network layer outputs. $\sigma(\cdot)$ is a non-linear activation function, like the ReLU (see [41]). A is a representative description of the graph structure in the form of an adjacency matrix (self-connection is included, see Section 2.2.2). D is the diagonal node degree matrix of A, and $W^{(l)}$ is a weight matrix for the i-th convolutional layer.

2.4. Accuracy Assessment

The proposed supervised and semi-supervised models will be trained on a series of training sets, while being evaluated on the test sets in chapter 3. Two widely used metrics are used for measuring the prediction precision and precision for each individual type:

$$prediction\ accuracy = \frac{\#(correct\ predictions)}{\#(total\ predictions)} \tag{9}$$

$$user\ accuracy_{type_i} = \frac{\#\left(correct\ predictions_{type_i}\right)}{\#\left(predictions_{type_i}\right)} \cdot 100\% \tag{10}$$

where $\#(\cdot)$ is the count of the corresponding predictions. As suggested by [50], the prediction accuracy is the fraction of the number of correct predictions over all the predictions. User accuracy is the probability that a value predicted to be in a certain class really is in that class. The probability is based on the fraction of correctly predicted values to the total number of values predicted to be in a class.

3. Results

For a model comparison and sensitivity test, the above-mentioned four supervised and one semi-supervised models are applied over the central urban region of Hong Kong (~91.13 km^2), for which locally complete data sets of urban functional land use are available and accessible. Specifically, the study region includes the Central and West, Wanchai, Eastern District, Kwun Tong, Kowloon City, Sham Shui Po and Yau Tsim Mong districts (see Figure 2a). After segmentation, this region is divided into 469 functional areas, which will be labeled and represented by feature vectors for further classification (for related histogram see Figure 2b). The frequency distribution of the logarithm area of the basic study units (functional areas) is plotted in Figure 2b; the mean (standard deviation) and median values are 0.139 (0.625) and 0.064 km^2, respectively. There is a total number of 171,704 POI points located in the study region. The related kernel density distribution shows hotspots in the economic central regions of Mongkok, Central district and Causeway Bay (see Figure 2c). Local urban functional land use is 100% (available from the LUHK dataset, Figure 2d), which provides an ideal study area for validating the candidates of the supervised and semi-supervised models. The total urban functional areas consist of 53.3% residential, 11.9% recreational, 17.1% commercial, 15.8% public service and 1.9% transportation areas (Figure 2d).

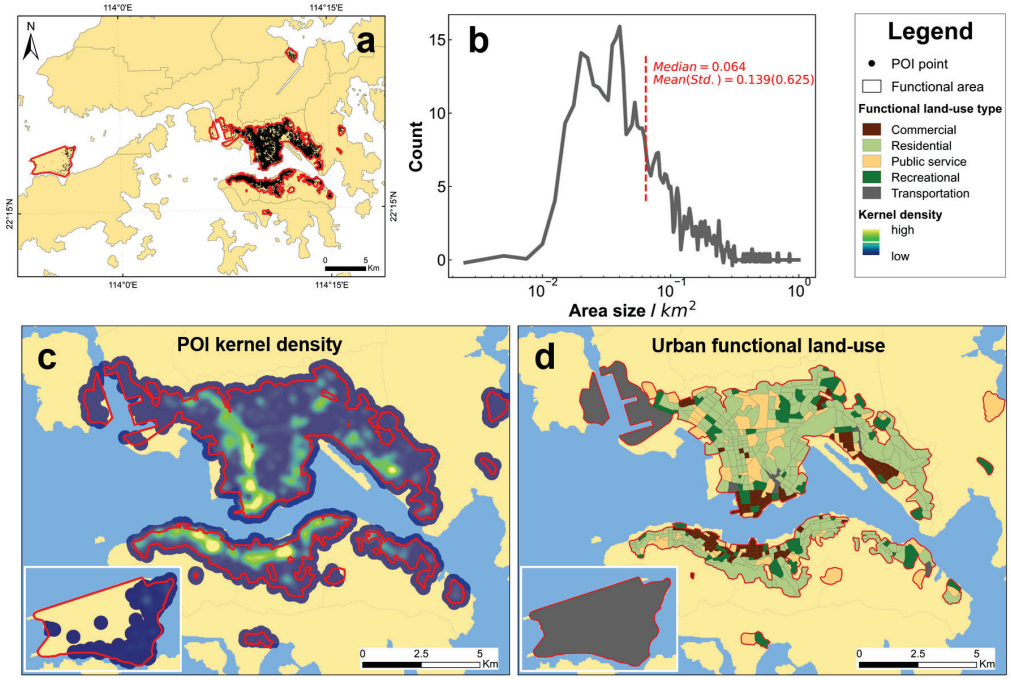


Figure 2. (**a**) Geographical setting of the study region, Hong Kong, China. The research is carried out within the red lines; (**b**) histogram statistics of the size for the functional area; (**c**) kernel density distribution of Point of Interest (POI) points; and (**d**) the functional land use distribution derived from the dataset of Land Utilization in Hong Kong (LUHK), including five categories: commercial, residential, public service, recreational and transportation.

3.1. Model Comparison

The five selected supervised and semi-supervised models are trained on a series of training sets with increasing number of training samples (ranging from 2% to 90% of the 469 functional areas). For the cases whose number of training samples are critically small (less than 5%), we manually select areas as training samples to ensure there are no types missing in the training set. Otherwise, the training/test sets are randomly split from all areas in the study region (see Appendix A, Table A2).

3.1.1. Sensitivity on Training Set Size

Figure 3a displays the accuracy changes along an increasing number of training samples, from which the following results are noted:

1. Although models' accuracies are improved as the amount of training data increases, disparities could be diagnosed from the model comparison. RF, MLP and GCN show an obvious higher accuracy and improving potential (see the tendency of the accuracies to the training sample percentage) as the number of training samples increases;
2. Supervised models (e.g., RF and MLP) indicate advantages with large number of training samples in terms of the accuracies. For example, MLP is with the top accuracies when the number of training samples is beyond around 10%. The RF-128 and RF-200 models show slightly improvements compared with MLP with the percentage of training samples equal to 30% and 40%. As the number of training samples keeps rising (greater than 40%), MLP wins again in terms of accuracy;
3. Semi-supervised models indicate advantages with a small number of training samples. That is, the GCN model is within the top accuracy, from 0.65 to 0.70, when the number of training samples is less than 10%. However, the potential of the GCN model is of underperformance when the number of training samples increases, compared with supervised models of RF and MLP;

4. All models are tested five times with the same amount and combination of training data for monitoring model stability by calculating the standard deviations (Std.) of the result accuracies (see the colored shadows in Figure 3a). SVM and GCN reveal better model stability (smaller Std.), compared with RF and MLP. As the model stability of MLP depends on the number of training data, it shows high instability with a small number of training data, but it improves (and become even better than RF) with a large number of training data.

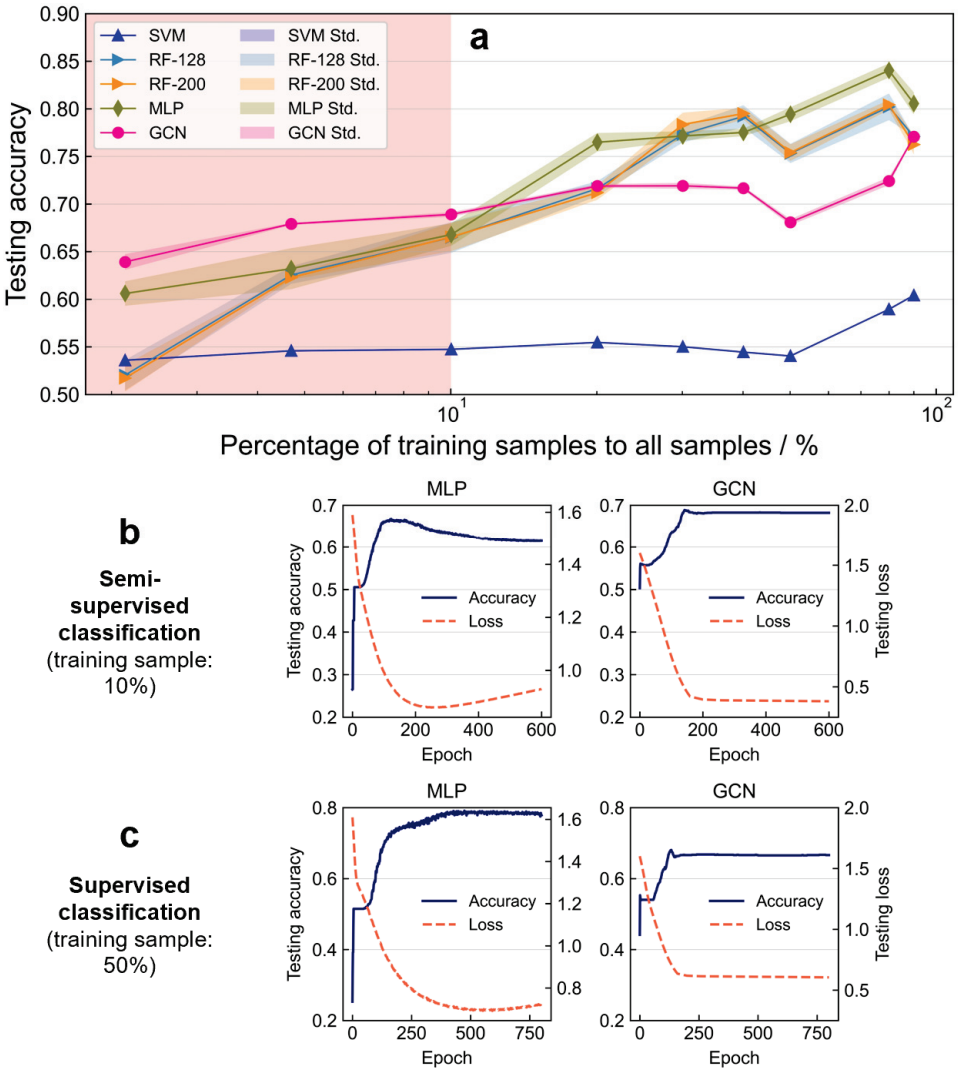


Figure 3. (**a**) Comparing training set size dependence associated with five functional area classification models: support vector machine (SVM), random forest (RF-128 and RF-200), multi-layer perceptron (MLP) and graph convolutional network (GCN), indicating a best performance with GCN under semi-supervised situation and a best performance with MLP under supervised situation; (**b**) and (**c**) the testing accuracy and testing loss curves for MLP and GCN during model trainings.

In general, RF and MLP (GCN) are better choices when presented with sufficient (insufficient) amounts of training data, while lacking training data (<10%) can significantly reduce the supervised model's accuracy.

3.1.2. Training Performances: Small vs. Large Number of Training Data

Different training performances under the same number of training sample conditions (here 10% and 50% are selected for comparison, and the prediction accuracy is the fraction of the number of correct predictions over the total number of samples) indicate (Figure 3b,c):

Small number of training data (10%): The best testing accuracy of MLP and GCN are similar with the number of epochs of ~200. However, GCN is better, because the testing accuracy of MLP first increases as the epoch increases but then decreases, which indicates an overfitting phenomenon: the production of an analysis that corresponds too closely or exactly to a particular set of data may, therefore, fail to fit to additional data or to predict future observations reliably [51].

Large number of training data (50%): GCN is approaching the highest accuracy around ~150 epochs, with which MLP indicates similar testing accuracy. However, the testing accuracy of MLP keeps improving as the number of training data increases and approaches 0.8, while GCN is facing a stagnation at less than 0.7.

3.1.3. Robustness to Different Selection of Training Data

To evaluate the performance of the 5 models, a further validation is conducted. 10 independent training sets, each consisting of 10% out of all samples, are generated, based on the stratified sampling [52] strategy. Another 10 training sets, each consisting of 50% out of all samples (one sample may occur in many sets), are generated similarly for a further cross-validation (Figure 4). The models are estimated on the test sets, which consist of the other 90% or 50% samples, to observe the stability and generalization ability [53,54]. The following results are noted:

Condition of small number of training data (10%): Accuracies for all the models (SVM, two RFs, MLP and GCN) are roughly divided into three levels: MLP and GCN are in the 1st class, RFs (with 128 and 200 decision trees) are in the 2nd class, and SVM should be the last choice in terms of accuracy. For MLP and GCN (the 1st class), GVN (MLP) shows a relatively higher (lower) accuracy and a relatively higher (lower) variability. For RFs (the 2nd class), more decision trees do not necessarily increase the accuracy, but may increase the variability.

Condition of large number of training data (50%): Apart from the SVM, all models yield higher accuracies and lower variabilities with rising training set sizes (Figure 4b). The RF-200 exhibits higher accuracies than the RF-128, while the two RFs still show the highest variabilities among all models. Two neural networks (MLP and GCN) show clear improvements on the robustness, whereas the accuracies of GCN are no more competitive. The MLP exhibits the highest accuracies of around 0.8, and a low sensitivity on the selection of training samples; therefore, it is preferred under the large number of training data conditions.

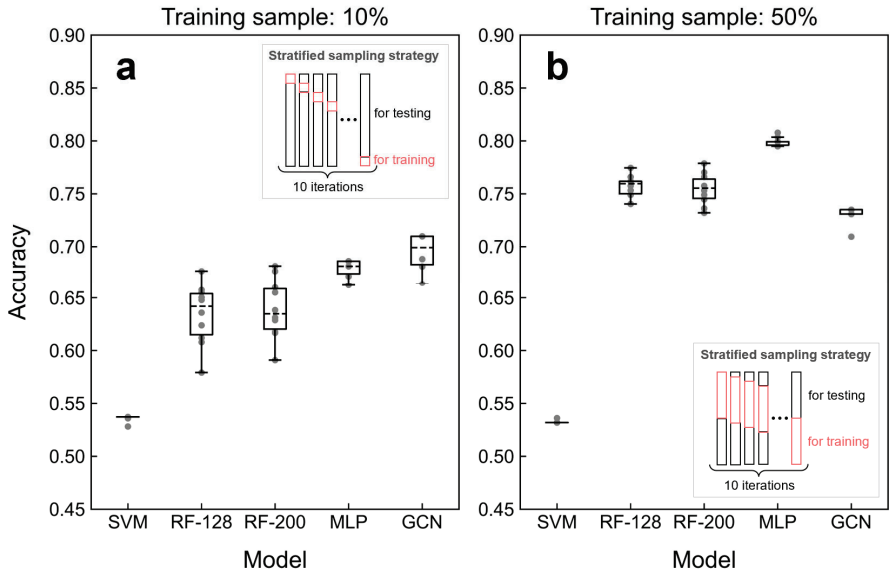


Figure 4. Cross validation results associated with five classification models (**a**) using small number of samples (10%) for training and (**b**) using large number of samples (50%) for training.

3.1.4. Impact from Different Levels of Functional Heterogeneity

To study the impact of the functional heterogeneity inside areas on the classification accuracy, 5 training sets, each consisting of 10%, are selected out of all samples (using the same stratified sampling method in Section 3.1.3) to train the GCN, and another 5, each consisting of 50% samples, are generated similarly to train the MLP. The occupation percentage of the labeled class (or purity) is calculated for the remaining 90% and 50% samples, based on the same method described in Section 2.1. The purity values are then segmented with an equal interval of 10% to form different purity ranks. The trained models are applied on those samples to observe the relationship between the accuracy and the purity rank. The results are plotted together with a frequency distribution of samples per purity rank in Figure 5.

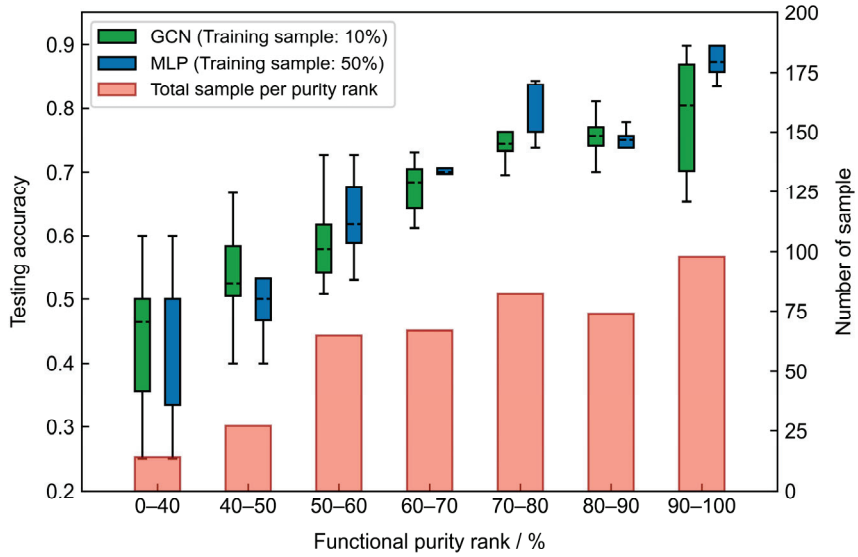


Figure 5. The correlation between accuracy and different functional purities with the frequency distribution of purity ranks.

Results reveal that, whether using 10% or 50% samples (to train GCN and MLP model, respectively), the classification accuracy increases with higher ranking of functional purity, and (or) larger number of samples per purity rank. On the one hand, mixed functions could blur the important features, making it more difficult for the model to distinguish the main function among compound functions. On the other hand, since there is an observed bias in the number of samples of different purities (fewer samples have low purity, as seen in Figure 5), the models may fail to learn enough knowledge to correctly classify low-purity samples. Therefore, we argue that the estimation of functional heterogeneity inside functional areas should be an important procedure in the entire workflow to indicate the confidence of classification results. For example, our experiment in Hong Kong showcases that using road network to divide and generate functional areas, the averaged purity is ~76.1%, and 90.4% samples have higher purity than 50%, meaning a reasonable overall confidence.

3.1.5. Visual Comparison

Figure 6 illustrates the details of urban functional area predictions and the comparisons with the reality maps (derived from the LUHK dataset of Hong Kong following the labeling methods in Section 2.1). In this comparison, to maximize model output differences, the number of training samples is set as 4.7% (known/unknown: 22/469) within the whole region. Images acquired from Sentinel-2A satellite with the spatial resolution of 10 × 10 m is used as a background. Given such a low quantity of training sample, the accuracy ranking from high to low is: GCN (0.68) > MLP (0.63) = RF-128 (0.63) > RF-200 (0.62) > SVM (0.55). As shown in the visual comparison, the GCN exhibits abilities to distinguish functional areas that other models fail to classify correctly. For example, the GCN model effectively

identified areas characterized by hospitals and schools (universities) (as indicated by the red boxes, located near Yau Ma Tei area), as well as major business and industrial areas (as indicated by yellow boxes, located at Kwun Tong district). In addition, despite introducing the one-vs.-rest strategy, the SVM classifier failed to distinguish multi-classes within the study area (Figure 6), which may be because it lacks the ability to handle the class imbalance problem introduced in Section 2.3.1.

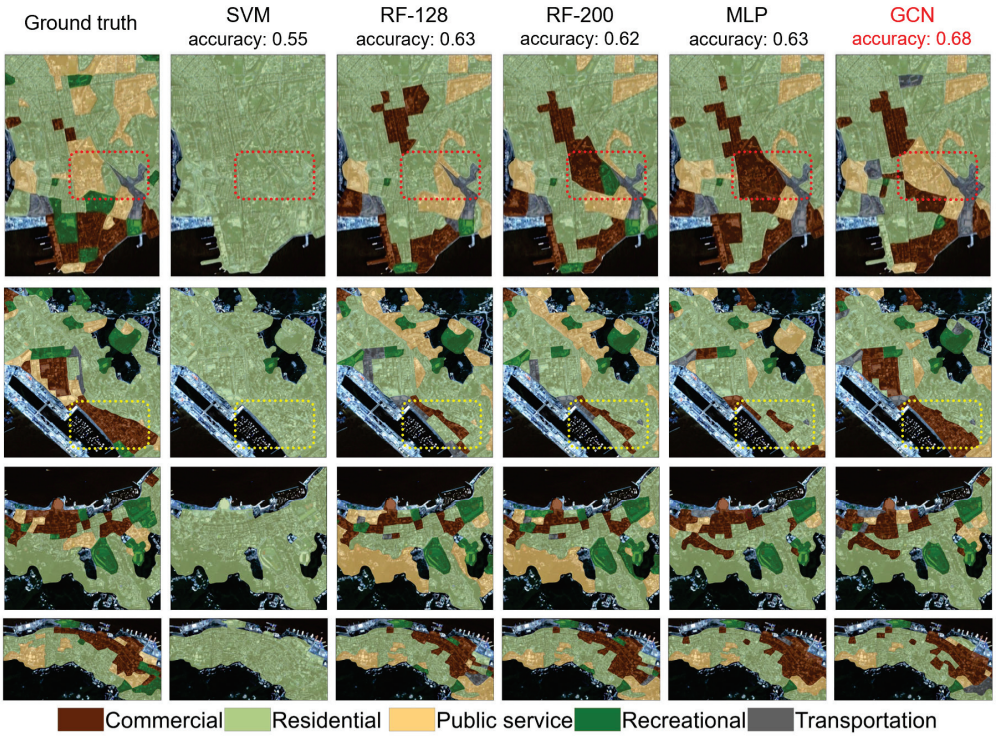


Figure 6. Visual comparison between ground truth and five functional land use prediction results obtained from SVM, RF-128, RF-200, MLP and GCN. The number of training samples is set as 4.7% (known/unknown: 22/469) of the total region, and the accuracy is highlighted accordingly.

3.2. Case Study: Beijing

The above analyses have compared and explained model sensitivity (on the training sample size), stability or reproducibility, accuracy, and robustness (from cross validation). Constraints on model applicability are comprehensively studied, which encouraged us to test the classification framework on a very different city in China. Unlike the relatively narrower roads, denser road network and smaller blocks observed in Hong Kong, the urban structure in Beijing is much less affected by terrain factors, but more by anthropological motivations [55,56].

The GCN is outstanding with a small number of training samples, which is important because, in real-world scenarios, lacking training samples of urban functional land use is one of the significant problems, especially in China. Here, for Beijing, the functional land use information of only ~20% of the central urban area (within the 5th ring, ~1140 km^2) is known (Figure 7). This small amount of training samples brings challenges and less accuracies if a supervised model (e.g., SVM, RF, MLP) is applied (see Section 3). Thus, the semi-supervised model of graph convolutional network (GCN) is selected in the following urban functional area prediction.

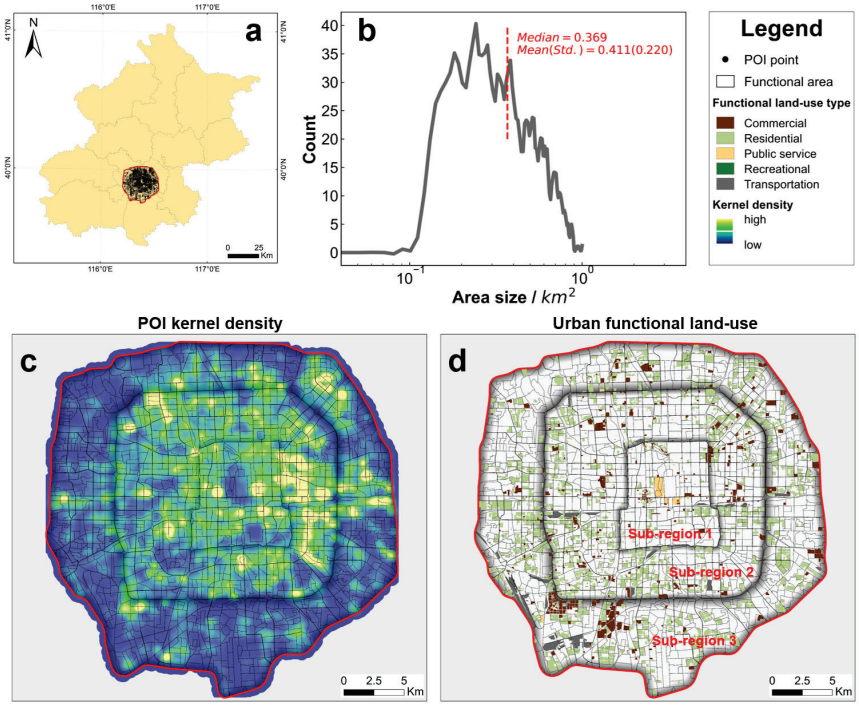


Figure 7. (**a**) Geographical setting of the study region, Beijing, China. The research is carried out within the 5th ring road (marked by the red line); (**b**) histogram statistics of the size for the functional area; (**c**) kernel density distribution of local point of interest (POI) points; and (**d**) the functional land use distribution obtained from the open street map land use dataset, including five categories: commercial, residential, public service, recreational and transportation.

Applications are carried out in parallel for three sub-regions of Beijing to promote the computing efficiency and highlight the heterogeneities within sub-regions. After data processing, 1571 functional areas are generated, with 9.4% (14/149), 8.81% (52/590) and 13.21% (110/833) of the three sub-regions (inside the 2nd ring, 2nd to 4th ring, and outside the 4th ring) selected as training samples. Three sub-regions cover different periods and degrees of development of the city. The 2nd ring encircles the famous Forbidden City and other archaeological landmarks, and outside the 4th ring the area shows less POI data (Figure 7b). Therefore, TF-IDF is calculated on the entire study region scale (within the 5th ring), while the semantic linkage of the cosine similarity is calculated on sub-region scale for highlighting the heterogeneities within sub-regions (see [57,58]).

GCN based classification and validation results are shown in Figure 8. The validation is achieved through field surveys and refers to the online map provided by the Amap platform close to the POI acquisition date. 20/149, 40/590 and 50/833 areas are randomly selected for validation, and the confusion matrixes of each sub-region are derived (Figure 8b). The following results are noticed:

5. Inside the 2nd ring, commercial, residential, public service, recreational and transportation areas are 51.0%, 8.1%, 35.6%, 4.7% and 0.7% of the related sub-region. From the 2nd to 4th ring, commercial, residential, public service, recreational and transportation areas are 62.7%, 3.1%, 19.7%, 13.9% and 0.7% of the related sub-region. And outside the 4th ring, commercial, residential, public service, recreational and transportation areas are 60.3%, 6.7%, 15.2%, 11.6% and 6% of the related sub-region;
6. The classification accuracies for three sub-regions are 0.60, 0.83 and 0.69, respectively, and 0.73 for the whole study area. The belt sub-region between the 2nd and 4th ring roads contains a complex distribution of various functional regions, yet exhibits the highest accuracy among three sub-regions. This is in agreement with the relatively high variability observed in the robustness test in Section 3.1.3;

7. Confusion matrixes indicate relatively higher user accuracies of residential (and recreational) functional land use from 64% to 88% (67% to 100%). The user accuracy of commercial (and public service) functional land use varies in three subareas from 40% to 80% (and 50% to 83%). Transportation functional land use is with less accuracy, which is probably because its low density;
8. More specifically, public service is located more in the north of the city, while transportation occurs more on the south. On the east side of the 2nd ring, Beijing Central Business District is clearly displaced (the clustered dark red colors). From the 2nd to 4th ring, the well-identified public service areas show the existence of corresponding institutes and universities, especially in the north. To the northwest part of the third sub-region, recreational areas, such as the Summer Palace and Yuanmingyuan, compose distinctive clusters (see the areas colored with dark green). POI point density is relatively low in this region, indicating a lower density of buildings and commercial activities. Areas in the south part of the city are more irregular in shape (Figure 8a), which may be explained by the fact that this region is relatively less planned and developed compared with other regions.

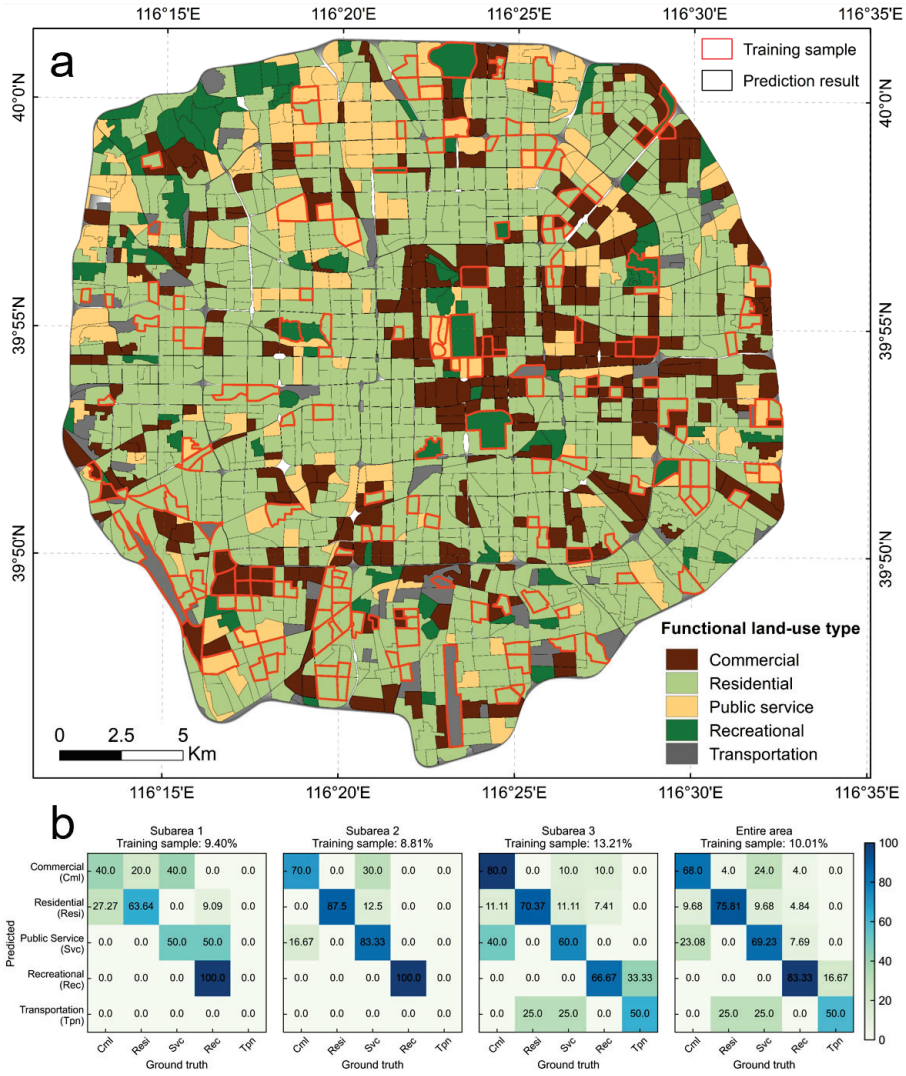


Figure 8. (**a**) Functional area prediction result within the 5th ring road of Beijing; (**b**) confusion matrixes of the 3 sub-regions, respectively. The row-normalized matrixes where diagonal element are user accuracies of the corresponding functional land use types.

4. Discussion

4.1. Sensitivity and Accuracy

The accuracy-efficiency tradeoff: Neural networks are complex architectures and require enormous amounts of training data with good quality to produce viable results. For one neural network, as the size of the training data grows, so does the output accuracies. This is in agreement with [59,60]. However, in choosing different supervised and semi-supervised models, it is not necessarily the more training data the better. An efficient training set size could be selected, according to the cost of the training data and the required output accuracy (Figure 3a). For example, with a 5% training sample, GCN reaches an accuracy of 0.68; with a 40% training sample, RF reaches an accuracy of 0.8; while, with a 80% training sample, MLP reaches an accuracy of 0.85. In this study, the semi-supervised GCN model has better performance in urban functional land use prediction, using only a small number (5%) of training samples, which is meaningful in large-scale, real-world applications.

Reproducibility: Lack of reproducibility in machine learning, which is a complex and growing issue exacerbated by a lack of code transparency, can affect safety, reliability and the detection of bias. In choosing supervised and semi-supervised models, reproducibility comes from two aspects: applying a model multiple times with the same training set (Figure 3a), and a different combination of the same amount of training data (see the cross validation in Figure 4). That is, to obtain a more reliable urban functional land use classification, a model with high reproducibility (less variable in the cross validation) is recommended. In this study, GCN is more stable with the same training set, but varies more in cross-validation with 10% training samples (Figure 4a). MLP is less stable with the same training set, but varies less in cross-validation. And for RF, the increasing of the number of decision tree may not increase the final accuracy, but may increase the model's instability. This is also reported in previous papers such as [61].

4.2. When Is Machine Learning Application the Best Choice?

As the size of a neural network's architecture grows, so does its requirement for the amounts of training data. Thus, with a large amount of training samples, model selection and validation would be easy. For example, in modifying present urban functional land use plans with keeping most of the original designs, supervised models such as RF and MLP could be selected. However, there is normally limited training data available in real-world applications. In such cases, supervised classifiers that once have performed well may fail, while exploiting the limited data by incorporating their sematic linkages will produce viable results; see, for example, with 5% training sample, GCN reaches an accuracy of 0.68. This is in agreement with previous research [27], which feeds both feature vectors and different linkages of functional areas (such as the origin and destination pairs of taxi trips) into a GCN model and obtains a lower error. As also implied in previous research [31], the forward and backward training processes of GCN are effectively equivalent to geographical weighted regression (GWR), which makes it suitable in understanding geographical phenomena.

4.3. Limitation and Future Work

Almost always labeled data is essential for machine learning (ML) models. Without enough high-quality labeled data, the use of ML is not recommended. Meanwhile, most ML algorithms work better when there is a spatial balanced or quantitatively equaled distribution for each urban functional area types. In addition, the overfitting problem is noticed when applying MLP over Hong Kong, but in the real application, there may not be enough samples for a validation set to monitor the training process and the overfitting problem. Thus, one should be cautious to apply MLP when there is less than 10% of the whole regions with known classification categories.

In addition to POI data used in this study, other geographical and remote sensing indexes could also improve the classification accuracy [62]. Other similarity methods could also be used to measure the linkage among urban functional areas (see the Moran's

I index and high-resolution image features for example [62,63]). In addition, different segmentation (e.g., OSM based in this study) and different study scale (e.g., sub-regions in Beijing) and urban functional characteristics in neighborhoods, communities, or even regions [37,62], result in varied prediction accuracy. Therefore, each specific method should be considered simultaneously with the available data source, the cost and efficiency for a better outcome. And for future work, within the basic study units, functional characteristics are actually mixed and require more detailed decomposition techniques to better represent the development intensities and interactions in a downscaled situation [27,39]. Furthermore, to deal with the current poor performance diagnosing low-purity samples (Figure 5), new techniques may be needed in the model training process, to specifically improve the ability to discriminate them

5. Conclusions

Formal land use refers to the qualitative attributes of land surface, while functional land use indicates its socioeconomic function. The formal land use map can be created with aerial or remote sensing images, but it is difficult to infer any functional attributes from these observations, especially for urban land use. City planners and other agencies have undertaken surveys to assign or infer the functional characteristics on basic urban areas under their jurisdiction [39]. Such an endeavor is often time-consuming, as the urban landscape is constantly changing with the construction/renovation of infrastructures, new commercial/residential/industrial developments, and the modification of existing uses [20,57,64]. Present geotagged data, e.g., point of interest (POI) data, bring new perspectives in data mining and supplement for defining urban functional area characteristics by associating with machine learning techniques. However, to our knowledge, there are still questions that need to be answered in using present geotagged data to diagnose urban functional areas, such as: When is a machine learning application the best choice? How to select a machine learning model? And what is the model's uncertainty, robustness and sensitivity?

Therefore, in this study, three supervised (SVM, RF, MLP) and one semi-supervised machine learning model (GCN) are selected. For both supervised and semi-supervised models, a normalized Term Frequency-Inverse Document Frequency (TF-IDF) feature vector for each functional area is calculated as model inputs for the three supervised (SVM, RF, MLP) models. Both the TF-IDF feature vector and the cosine similarity of the TF-IDF feature vectors from two different areas are calculated as model inputs for the semi-supervised GCN model. Followed by the uncertainty, robustness and sensitivity tests (see Section 3), the following results are noticed:

1. As the amount of training sample grows, models' accuracies are improved, but with different potentials. GCN model is with the top accuracy, from 0.65 to 0.70, when the number of training samples is less than 10%, while MLP and RF show top accuracies when the number of training samples exceeds around 10%;
2. With a large amount of training samples, which is normally in the modification of existing urban functional area maps, RF and MLP could be the best selection. However, one should note that MLP is less stable with the same training set, but varies less in cross-validation. For RF, the increasing of the number of decision trees may not increase the final accuracy, but may increase the model's instability;
3. With a small amount of training samples, which is normally the case in the real world, GCN could provide viable results by incorporating the auxiliary information provided by the proposed semantic linkages. For example, with the incorporating of the similarity-based semantic linkage, the model could be trained using only 5% of the total samples and produce an accuracy of 0.68;
4. In the perspective of the model overfit problem, which could be ignored in the real application due to lacking enough testing samples, when the training samples is less than 10%, we suggest choosing GCN for the urban functional land use prediction, and one should be cautious using MLP, by testing the optimal epoch for obtaining the best accuracy.

Author Contributions: Conceptualization, R.D., Y.G., D.C. and T.Y.; Formal analysis, R.D.; Funding acquisition, Y.G. and S.G.; Methodology, R.D.; Project administration, Y.G. and S.G.; Resources, Z.L., Z.W. and S.G.; Supervision, Y.G. and S.G.; Validation, C.Z. and J.T.; Visualization, R.D.; Writing—original draft, R.D.; Writing—review and editing, R.D., D.C., K.F. and T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Strategic Priority Research Program of Chinese Academy of Sciences, grant number XDA23100504, the National Key Research and Development Program of China, grant number 2019YFD1100705 and the Research on the Beijing central axis landscape view gallery and city skyline.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [https://gitee.com/pickup20/multi-modal-paper/tree/mastere/data]. Data accessed during March and April 2022.

Acknowledgments: Supported by the Strategic Priority Research Program of Chinese Academy of Sciences (XDA23100504), the National Key Research and Development Program of China (2019YFD1100705) and the Research on the Beijing central axis landscape view gallery and city skyline. The authors appreciate the valuable comments from the editor and three reviewers.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A

Table A1. POI categories used in this paper.

ID	POI Category	ID	POI Category
0	Automobile Service Related	135	Stationary Store
1	Filling Station	136	Sports Store
2	Other Energy Station	137	Commercial Street
3	Automobile Maintenance/Decoration	138	Clothing Store
4	Car Wash	139	Franchise Store
5	Automobile Club	140	Special Trade House
6	Automobile Rescue	141	Personal Care Items Shop
7	Automobile Parts Sales	142	Daily Life Service Place
8	Automobile Rental	143	Travel Agency
9	Used Automobile Dealer	144	Information Centre
10	Charging Station	145	Ticket Office
11	Automobile Sales	146	Post Office
12	Volkswagen Franchised Sales	147	Post Office
13	Honda Franchised Sales	148	Logistics Service
14	Audi Franchised Sales	149	Telecom Office
15	General Motors Franchised Sales	150	Professional Service Firm
16	BMW Franchised Sales	151	Job Center
17	Nissan Franchised Sales	152	Water Supply Service Office

Table A1. *Cont.*

ID	POI Category	ID	POI Category
18	Renault Franchised Sales	153	Electric Supply Service Office
19	Mercedes-Benz Franchised Sales	154	Beauty and Hairdressing Store
20	Toyota Franchised Sales	155	Repair Store
21	Subaru Franchised Sales	156	Photo Finishing
22	Peugeot Citroen Franchised Sales	157	Bath & Massage Center
23	Peugeot Citroen	158	Laundry
24	Mitsubishi Franchised Sales	159	Agency
25	Fiat Franchised Sales	160	Move Service
26	Ferrari Franchised Sales	161	Lottery Store
27	Hyundai Franchised Sales	162	Funeral Facilities
28	KIA Franchised Sales	163	Baby Service Place
29	Ford Franchised Sales	164	Shared Device
30	JAGUAR Franchised Sales	165	Sports & Recreation Places
31	Land Rover Franchised Sales	166	Sports Stadium
32	Porsche Franchised Sales	167	Golf Related
33	DFM Franchised Sales	168	Recreation Center
34	Geely Franchised Sales	169	Holiday & Nursing Resort
35	Chery Franchised Sales	170	Recreation Place
36	Chrysler Franchised Sales	171	Theatre & Cinema
37	ROEWE Sales	172	Medical and Health Care Service Place
38	MG Sales	173	Hospital
39	JAC Sales	174	Special Hospital
40	Hongqi Sales	175	Clinic
41	Chang'an Sales	176	Emergency Center
42	Haima Sales	177	Disease Prevention Institution
43	BAIC MOTOR Sales	178	Pharmacy
44	Great Wall Sales	179	Veterinary Hospital
45	Luxgen Sales	180	Accommodation Service Related
46	GAC Trumpchi Sales	181	Hotel
47	Truck Sales	182	Hostel
48	Dongfeng Truck Sales	183	Tourist Attraction Related
49	SINOTRUK Sales	184	Park & Square
50	FAW Jiefang Sales	185	Park & Plaza
51	Foton Truck Sales	186	Scenery Spot
52	Shaanxi Heavy-duty Truck Sales	187	Commercial House Related
53	Beiben Trucks Sales	188	Industrial Park
54	JAC Truck Sales	189	Building
55	CAMC Sales	190	Residential Area
56	Chengdu Dayun Automotive Sales	191	Governmental & Social Groups Related
57	Mercedes-Benz Truck Sales	192	Governmental Organization
58	MAN Sales	193	Foreign Organization

Table A1. *Cont.*

ID	POI Category	ID	POI Category
59	SCANIA Sales	194	Democratic Party
60	Volvo Truck Sales	195	Social Group
61	Qoros Sales	196	Public Security Organization
62	Automobile Repair	197	Traffic Vehicle Management
63	Automobile Comprehensive Repair	198	Industrial and Commercial Taxation Institution
64	Volkswagen Franchised Repair	199	Science & Education Cultural Place
65	Honda Franchised Repair	200	Museum
66	Audi Franchised Repair	201	Exhibition Hall
67	General Motors Franchised Repair	202	Convention & Exhibition Center
68	BMW Franchised Repair	203	Art Gallery
69	Nissan Franchised Repair	204	Library
70	Renault Franchised Repair	205	Science & Technology Museum
71	Mercedes-Benz Franchised Repair	206	Planetarium
72	Toyota Franchised Repair	207	Cultural Palace
73	Subaru Franchised Repair	208	Archives Hall
74	Peugeot Citroen Franchised Repair	209	Arts Organization
75	Peugeot Citroen	210	Media Organization
76	Mitsubishi Franchised Repair	211	School
77	Fiat Franchised Repair	212	Research Institution
78	Ferrari Franchised Repair	213	Training Institution
79	Hyundai Franchised Repair	214	Driving School
80	KIA Franchised Repair	215	Transportation Service Related
81	Ford Franchised Repair	216	Airport Related
82	JAGUAR Franchised Repair	217	Railway Station
83	Land Rover Franchised Repair	218	Port & Marina
84	Porsche Franchised Repair	219	Coach Station
85	DFM Franchised Repair	220	Subway Station
86	Geely Franchised Repair	221	Light Rail Station
87	Chery Franchised Repair	222	Bus Station
88	Chrysler Franchised Repair	223	Commuter Bus Station
89	ROEWE Repair	224	Parking Lot
90	MG Repair	225	Border Crossing
91	JAC Repair	226	Taxi
92	Hongqi Repair	227	Ferry Station
93	Chang'an Repair	228	Ropeway Station
94	Haima Repair	229	Loading & Unloading Area
95	BAIC MOTOR Repair	230	Finance & Insurance Service Institution
96	Great Wall Repair	231	Bank
97	Luxgen Repair	232	Bank Related
98	GAC Trumpchi Repair	233	ATM
99	Truck Repair	234	Insurance Company

Table A1. *Cont.*

100	Dongfeng Truck Repair	235	Securities Company
101	SINOTRUK Repair	236	Finance Company
102	FAW Jiefang Repair	237	Enterprises
103	Foton Truck Repair	238	Famous Enterprise
104	Shaanxi Heavy-duty Truck Repair	239	Company
105	Beiben Trucks Repair	240	Factory
106	JAC Truck Repair	241	Farming, Forestry, Animal Husbandry and Fishery Base
107	CAMC Repair	242	Road Furniture
108	Chengdu Dayun Automotive Repair	243	Warning Sign
109	Mercedes-Benz Truck Repair	244	Toll Gate
110	MAN Repair	245	Service Area
111	SCANIA Repair	246	Traffic Light
112	Volvo Truck Repair	247	Signpost
113	Qoros Repair	248	Place Name & Address
114	Motorcycle Service Related	249	Natural Place Name
115	Motorcycle Sales	250	Transportation Place Name
116	Motorcycle Repair	251	Address Sign
117	Food & Beverages Related	252	City Center
118	Chinese Food Restaurant	253	Landmark Buildings
119	Foreign Food Restaurant	254	The hot names
120	Fast Food Restaurant	255	Public Facility
121	Leisure Food Restaurant	256	Newsstand
122	Coffee House	257	Public Phone
123	Tea House	258	Public Toilet
124	Icecream Shop	259	Emergency Shelter
125	Bakery	260	Incidents and Events
126	Dessert House	261	Public Event
127	Shopping Related Places	262	Emergency
128	Shopping Plaza	263	Indoor facilities
129	Convenience Store	264	Pass Facilities
130	Home Electronics Hypermarket	265	Gate of Buildings
131	Supermarket	266	Gate of Street House
132	Plants & Pet Market	267	Virtual Gate
133	Home Building Materials Market	268	Special corridor
134	Comprehensive Market		

Table A2. Statistics of the training sets used for model comparison experiment.

Percentage of Raining Sample (%)	Area Type	Number of Samples	Percentage of Raining Sample (%)	Area Type	Number of Samples
2	Commercial	1	40	Commercial	34
	Residential	3		Residential	97
	Public service	3		Public service	30
	Recreational	2		Recreational	23
	Transportation	1		Transportation	4
4.7	Commercial	6	50	Commercial	42
	Residential	4		Residential	123
	Public service	4		Public service	32
	Recreational	4		Recreational	32
	Transportation	4		Transportation	5
10	Commercial	8	80	Commercial	65
	Residential	19		Residential	196
	Public service	8		Public service	62
	Recreational	8		Recreational	46
	Transportation	4		Transportation	5
20	Commercial	25	90	Commercial	79
	Residential	42		Residential	222
	Public service	19		Public service	63
	Recreational	6		Recreational	48
	Transportation	2		Transportation	9
30	Commercial	24			
	Residential	69			
	Public service	25			
	Recreational	20			
	Transportation	2			

References

1. Duranton, G.; Puga, D. Chapter 8—Urban Land Use. In *Handbook of Regional and Urban Economics*; Duranton, G., Henderson, J.V., Strange, W.C., Eds.; Elsevier: Amsterdam, The Netherlands, 2015; pp. 467–560.
2. Cai, D.; Fraedrich, K.; Guan, Y.; Guo, S.; Zhang, C. Urbanization and the thermal environment of Chinese and US-American cities. *Sci. Total Environ.* **2017**, *589*, 200–211. [CrossRef] [PubMed]
3. Xing, Q.; Sun, Z.; Tao, Y.; Shang, J.; Miao, S.; Xiao, C.; Zheng, C. Projections of future temperature-related cardiovascular mortality under climate change, urbanization and population aging in Beijing, China. *Environ. Int.* **2022**, *163*, 107231. [CrossRef] [PubMed]
4. Asabere, S.B.; Acheampong, R.A.; Ashiagbor, G.; Beckers, S.C.; Keck, M.; Erasmi, S.; Schanze, J.; Sauer, D. Urbanization, land use transformation and spatio-environmental impacts: Analyses of trends and implications in major metropolitan regions of Ghana. *Land Use Policy* **2020**, *96*, 104707. [CrossRef]
5. Dadashpoor, H.; Azizi, P.; Moghadasi, M. Land use change, urbanization, and change in landscape pattern in a metropolitan area. *Sci. Total Environ.* **2019**, *655*, 707–719. [CrossRef] [PubMed]
6. Hu, S.; He, Z.; Wu, L.; Yin, L.; Cui, H. A framework for extracting urban functional regions based on multiprototype word. *Computers Environ. Urban Syst.* **2019**, *80*, 101442. [CrossRef]
7. Goldstein, J.H.; Caldarone, G.; Duarte, T.K.; Ennaanay, D.; Hannahs, N.; Mendoza, G.; Polasky, S.; Wolny, S.; Daily, G.C. Integrating ecosystem-service tradeoffs into land-use decisions. *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 7565–7570. [CrossRef]
8. Frias-Martinez, V.; Frias-Martinez, E. Spectral clustering for sensing urban land use using Twitter activity. *Eng. Appl. Artif. Intell.* **2014**, *35*, 237–245. [CrossRef]
9. Zhai, W.; Bai, X.; Shi, Y.; Han, Y.; Peng, Z.R.; Gu, C. Beyond Word2vec: An approach for urban functional region extraction and identification by combining Place2vec and POIs. *Comput. Environ. Urban Syst.* **2019**, *74*, 1–12. [CrossRef]
10. Gao, S.; Janowicz, K.; Couclelis, H. Extracting urban functional regions from points of interest and human activities on location-based social networks. *Trans. GIS* **2017**, *21*, 446–467. [CrossRef]
11. Sanchez, T.W.; Shumway, H.; Gordner, T.; Lim, T. The prospects of artificial intelligence in urban planning. *Int. J. Urban Sci.* **2022**, 1–16. [CrossRef]
12. Zhang, J.; He, X.; Yuan, X.-D. Research on the relationship between Urban economic development level and urban spatial structure—A case study of two Chinese cities. *PLoS ONE* **2020**, *15*, e0235858. [CrossRef] [PubMed]
13. Du, M.; Zhang, X.; Mora, L. Strategic Planning for Smart City Development: Assessing Spatial Inequalities in the Basic Service Provision of Metropolitan Cities. *J. Urban Technol.* **2021**, *28*, 115–134. [CrossRef]
14. Cariolet, J.-M.; Colombert, M.; Vuillet, M.; Diab, Y. Assessing the resilience of urban areas to traffic-related air pollution: Application in Greater Paris. *Sci. Total Environ.* **2018**, *615*, 588–596. [CrossRef] [PubMed]

15. Hao, H.; Wang, Y. Disentangling relations between urban form and urban accessibility for resilience to extreme weather and climate events. *Landsc. Urban Plan.* **2022**, *220*, 104352. [CrossRef]
16. Kim, D.; Song, S.-K. Measuring changes in urban functional capacity for climate resilience: Perspectives from Korea. *Futures* **2018**, *102*, 89–103. [CrossRef]
17. Ouyang, X.; Wei, X.; Li, Y.; Wang, X.-C.; Klemeš, J.J. Impacts of urban land morphology on PM2. 5 concentration in the urban agglomerations of China. *J. Environ. Manag.* **2021**, *283*, 112000. [CrossRef]
18. Yu, Z.; Jing, Y.; Yang, G.; Sun, R. A new urban functional zone-based climate zoning system for urban temperature study. *Remote Sens.* **2021**, *13*, 251. [CrossRef]
19. Yao, Y.; Liu, P.; Hong, Y.; Liang, Z.; Wang, R.; Guan, Q.; Chen, J. Fine-scale intra- and inter-city commercial store site recommendations using knowledge transfer. *Trans. GIS* **2019**, *23*, 1029–1047. [CrossRef]
20. Klapka, P.; Kraft, S.; Halás, M. Network based definition of functional regions: A graph theory approach for spatial distribution of traffic flows. *J. Transp. Geogr.* **2020**, *88*, 102855. [CrossRef]
21. Zhao, P.; Luo, A.; Liu, Y.; Xu, J.; Li, Z.; Zhuang, F.; Sheng, V.S.; Zhou, X. Where to Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 2512–2524. [CrossRef]
22. Shen, G.; Zhao, Z.; Kong, X. GCN2CDD: A Commercial District Discovery Framework via Embedding Space Clustering on Graph Convolution Networks. *IEEE Trans. Ind. Inform.* **2022**, *18*, 356–364. [CrossRef]
23. Yang, J.; Wang, Y.; Xiu, C.; Xiao, X.; Xia, J.; Jin, C. Optimizing local climate zones to mitigate urban heat island effect in human settlements. *J. Clean. Prod.* **2020**, *275*, 123767. [CrossRef]
24. Lyu, Y.; Wang, M.; Zou, Y.; Wu, C. Mapping trade-offs among urban fringe land use functions to accurately support spatial planning. *Sci. Total Environ.* **2022**, *802*, 149915. [CrossRef]
25. Guan, Q.; Zhou, J.; Wang, R.; Yao, Y.; Qian, C.; Zhai, Y.; Ren, S. Understanding China's urban functional patterns at the county scale by using time-series social media data. *J. Spat. Sci.* **2022**, 1–19. [CrossRef]
26. Cai, M. Natural language processing for urban research: A systematic review. *Heliyon* **2021**, *7*, e06322. [CrossRef] [PubMed]
27. Zhu, D.; Zhang, F.; Wang, S.; Wang, Y.; Cheng, X.; Huang, Z.; Liu, Y. Understanding Place Characteristics in Geographic Contexts through Graph Convolutional Neural Networks. *Ann. Am. Assoc. of Geogr.* **2020**, *110*, 408–420. [CrossRef]
28. Xu, N.; Luo, J.; Wu, T.; Dong, W.; Liu, W.; Zhou, N. Identification and portrait of urban functional zones based on multisource heterogeneous data and ensemble learning. *Remote Sens.* **2021**, *13*, 373. [CrossRef]
29. Zhang, W.; Liu, H.; Liu, Y.; Zhou, J.; Xu, T.; Xiong, H. Semi-Supervised City-Wide Parking Availability Prediction via Hierarchical Recurrent Graph Neural Network. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 3984–3996. [CrossRef]
30. Kim, N.; Yoon, Y. Effective Urban Region Representation Learning Using Heterogeneous Urban Graph Attention Network (HUGAT). *arXiv* **2022**, arXiv:2202.09021.
31. Zhu, D.; Liu, Y.; Yao, X.; Fischer, M.M. Spatial regression graph convolutional neural networks: A deep learning paradigm for spatial multivariate distributions. *GeoInformatica* **2022**, *26*, 645–676. [CrossRef]
32. Geng, X.; Li, Y.; Wang, L.; Zhang, L.; Yang, Q.; Ye, J.; Liu, Y. Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 3656–3663. [CrossRef]
33. Chi, J.; Jiao, L.; Dong, T.; Gu, Y.; Ma, Y. Quantitative identification and visualization of urban functional area based on POI data. *J. Geomat* **2016**, *41*, 68–73.
34. Miao, R.; Wang, Y.; Li, S. Analyzing urban spatial patterns and functional zones using sina Weibo POI data: A case study of Beijing. *Sustainability* **2021**, *13*, 647. [CrossRef]
35. Li, Y.; Liu, C.; Li, Y. Identification of Urban Functional Areas and Their Mixing Degree Using Point of Interest Analyses. *Land* **2022**, *11*, 996. [CrossRef]
36. Zhang, X.; Du, S.; Wang, Q. Hierarchical semantic cognition for urban functional zones with VHR satellite images and POI data. *ISPRS J. Photogramm. Remote Sens.* **2017**, *132*, 170–184. [CrossRef]
37. Zhan, X.; Ukkusuri, S.V.; Zhu, F. Inferring Urban Land Use Using Large-Scale Social Media Check-in Data. *Netw. Spat. Econ.* **2014**, *14*, 647–667. [CrossRef]
38. Wong, K.S.; Tanaka, K. Data embedding for geo-tagging any contents in smart device. In Proceedings of the 2014 IEEE Region 10 Symposium, Kuala Lumpur, Malaysia, 14–16 April 2014.
39. Hu, Y.; Han, Y. Identification of Urban Functional Areas Based on POI Data: A Case Study of the Guangzhou Economic and Technological Development Zone. *Sustainability* **2019**, *11*, 1385. [CrossRef]
40. Iranmanesh, A.; Cömert, N.Z.; Hoşkara, Ş.Ö. Reading urban land use through spatio-temporal and content analysis of geotagged Twitter data. *GeoJournal* **2022**, *87*, 2593–2610. [CrossRef]
41. Kipf, T.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
42. Yuan, J.; Zheng, Y.; Xie, X. Discovering regions of different functions in a city using human mobility and POIs. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining 2012, Association for Computing Machinery, Beijing, China, 12–16 August 2012; pp. 186–194.
43. Liu, W.; Lü, L. Link prediction based on local random walk. *EPL (Europhys. Lett.)* **2010**, *89*, 58007. [CrossRef]
44. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
45. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]

46. Saravanan, R.; Sujatha, P. A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification. In Proceedings of the 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 14–15 June 2018.
47. Aly, M. Survey on multiclass classification methods. *Neural Netw* **2005**, *19*, 9.
48. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *99*, 2999–3007.
49. Woldesemayat, E.M.; Genovese, P.V. Urban Green Space Composition and Configuration in Functional Land Use Areas in Addis Ababa, Ethiopia, and Their Relationship with Urban Form. *Land* **2021**, *10*, 85. [CrossRef]
50. Junker, M.; Hoch, R.; Dengel, A. On the evaluation of document analysis components by recall, precision, and accuracy. In Proceedings of the Fifth International Conference on Document Analysis and Recognition ICDAR'99 (Cat. No. PR00318), Bangalore, India, 22 September 1999.
51. Dietterich, T. Overfitting and undercomputing in machine learning. *ACM Comput. Surv. (CSUR)* **1995**, *27*, 326–327. [CrossRef]
52. Parsons, V.L. Stratified sampling. *Wiley StatsRef Stat. Ref. Online* **2014**, 1–11.
53. Stone, M. Cross-validation:a review. *Ser. Stat.* **1978**, *9*, 127–139. [CrossRef]
54. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful are Graph Neural Networks? *arXiv* **2019**, arXiv:1810.00826.
55. Ganesan, S.; Lau, S.S.Y. Urban challenges in Hong Kong: Future directions for design. *Urban Des. Int.* **2000**, *5*, 3–12. [CrossRef]
56. Liu, G.; Yang, Z.; Chen, B.; Ulgiati, S. Monitoring trends of urban development and environmental impact of Beijing, 1999–2006. *Sci. Total Environ.* **2011**, *409*, 3295–3308. [CrossRef] [PubMed]
57. Zhang, X.; Du, S. A Linear Dirichlet Mixture Model for decomposing scenes: Application to analyzing urban functional zonings. *Remote Sens. Environ.* **2015**, *169*, 37–49. [CrossRef]
58. Wang, J.-F.; Zhang, T.-L.; Fu, B.-J. A measure of spatial stratified heterogeneity. *Ecol. Indic.* **2016**, *67*, 250–256. [CrossRef]
59. van Engelen, J.E.; Hoos, H.H. A survey on semi-supervised learning. *Mach. Learn.* **2020**, *109*, 373–440. [CrossRef]
60. Li, C.; Wang, J.; Wang, L.; Hu, L.; Gong, P. Comparison of Classification Algorithms and Training Sample Sizes in Urban Land Classification with Landsat Thematic Mapper Imagery. *Remote Sens.* **2014**, *6*, 964–983. [CrossRef]
61. Jin, H.; Stehman, S.V.; Mountrakis, G. Assessing the impact of training sample selection on accuracy of an urban classification: A case study in Denver, Colorado. *Int. J. Remote Sens.* **2014**, *35*, 2067–2081. [CrossRef]
62. Zhang, K.; Ming, D.; Du, S.; Xu, L.; Ling, X.; Zeng, B.; Lv, X. Distance Weight-Graph Attention Model-Based High-Resolution Remote Sensing Urban Functional Zone Identification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–18. [CrossRef]
63. Wu, Y.; Zhuang, D.; Labbe, A.; Sun, L. Inductive Graph Neural Networks for Spatiotemporal Kriging. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 4478–4485. [CrossRef]
64. Wang, Y.; Wang, T.; Tsou, M.-H.; Li, H.; Jiang, W.; Guo, F. Mapping Dynamic Urban Land Use Patterns with Crowdsourced Geo-Tagged Social Media (Sina-Weibo) and Commercial Points of Interest Collections in Beijing, China. *Sustainability* **2016**, *8*, 1202. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

remote sensing

MDPI

Article

The Effect of Negative Samples on the Accuracy of Water Body Extraction Using Deep Learning Networks

Jia Song [1,2,*] and Xiangbing Yan [1,3]

[1] State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China
[2] Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China
[3] School of Resource and Environmental Science, Wuhan University, Wuhan 430072, China
* Correspondence: songj@igsnrr.ac.cn

Abstract: Water resources are important strategic resources related to human survival and development. Water body extraction from remote sensing images is a very important research topic for the monitoring of global and regional surface water changes. Deep learning networks are one of the most effective approaches and training data is indispensable for ensuring the network accurately extracts water bodies. The training data for water body extraction includes water body samples and non-water negative samples. Cloud shadows are essential negative samples due to the high similarity between water bodies and cloud shadows, but few studies quantitatively evaluate the impact of cloud shadow samples on the accuracy of water body extraction. Therefore, the training datasets with different proportions of cloud shadows were produced, and each of them includes two types of cloud shadow samples: the manually-labeled cloud shadows and unlabeled cloud shadows. The training datasets are applied on a novel transformer-based water body extraction network to investigate how the negative samples affect the accuracy of the water body extraction network. The evaluation results of Overall Accuracy (OA) of 0.9973, mean Intersection over Union (mIoU) of 0.9753, and Kappa of 0.9747 were obtained, and it was found that when the training dataset contains a certain proportion of cloud shadows, the trained network can handle the misclassification of cloud shadows well and more accurately extract water bodies.

Keywords: water body extraction; deep learning; negative sample; cloud shadow interference; semantic segmentation

1. Introduction

Water resources play an important role in nature, providing necessary resources for human life, industry production and agriculture planting. Since remote sensing satellites have the advantages of large-scale detection of land surface, many remote sensing images have been applied in water body extraction and monitoring the change in water resources [1,2]. Based on the remote sensing images, various methods [3–6] are developed to extract water bodies from remote sensing images, and deep learning is the most popular approach among them [7–10].

The deep learning approach shows great potential in large-scale automatic image classification tasks due to its strong generalization ability. The main deep learning networks used in water body extraction are semantic segmentation networks. FCN [11–14], U-Net [15–19], DeepLab [20–22] and SegNet [23–25] are commonly used semantic segmentation networks to extract water bodies. In addition, some studies focus on the improvement based on them and achieve better accuracy of water body extraction [26–30]. While training samples also play an important role on accuracy improvement, since the deep learning is a data-driven approach and the accuracy heavily replies on the training samples. The existing efforts on training samples include replacing the low-resolution images with the higher-resolution

images to achieve fine water body extraction [31–33] and the utilization of data augmentation techniques [34–36] to compensate for insufficient samples. The main attention of these studies is paid to positive samples for improving the precision of water bodies.

However, the training samples should include not only positive samples but also negative samples. Negative samples in water body extraction refers to the samples that contains non-water objects. With the negative samples, the deep learning networks may better learn the discriminative features of water bodies. This cognition is formed based on the study [37], which proposes a concept of deep feature space for classification tasks. Based on the deep feature space, a gap between the per-class distributions of the training and testing vectors is accurately described. The bigger gap means higher separability of different classes, which helps classifiers to better distinguish different classes. The negative samples may be beneficial for forming the high separability of per-class distributions, thereby obtaining high accuracy.

In water body extraction, cloud shadows can be indispensable negative samples since cloud shadows are easily misclassified as water bodies due to the similarity between them. The existing efforts have still not well addressed this issue. Instead, they usually use cloud-free images or utilize extra cloud-shadow masks in water body extraction [38,39]. The cloud-free images are obtained by directly selecting non-cloud images or by using image composition algorithms. This approach is obviously limited by the situation of the cloud coverage and is not suitable for the areas that are often covered by clouds. When the networks are trained with cloud-free samples, they cannot well distinguish the water bodies and cloud shadows, which leads to the misclassification of cloud shadows. The approach to utilize cloud-shadow masks for eliminating cloud shadows is less efficient since it requires two stages of mask generation and water body extraction. Moreover, the accuracy of cloud shadow masks obtained by Fmask [40] or Tmask [41] algorithms is not very high; thus, they may not effectively avoid the misclassification of cloud shadows.

Therefore, considering that the misclassification of cloud shadows is not well addressed and negative samples in deep learning are not paid significant attention, this study investigates the impact of cloud shadow negative samples on the accuracy of water body extraction, especially for the low accuracy raised by the cloud shadows being misclassified as waters. Moreover, the impact of negative samples on the precision of water boundaries and the water bodies being missed are also involved. We thus designed two groups of training data. Both these groups are based on the same set of Sentinel-2 images, but the difference of the two groups is the labeled aspects. In one group, only water bodies are labeled, and in the other group, both water bodies and cloud shadows are labeled. Each group of the training data further consists of two training datasets with different proportions of cloud shadows, and these training datasets, respectively, are used to train a novel vison transformer-based deep learning network for extracting water bodies. By analyzing the extraction results, we can find how the cloud-shadow negative samples affect the accuracy of water body extraction.

2. Materials and Methods

2.1. Study Area

The western Tibetan Plateau is the study area in this work (Figure 1). It is located at 82.0° E to 92.3° E and 30.3° N to 36.3° N. There are approximately 1500 lakes with an area greater than 1 km^2 in this region [42]. The region affects the global environment and is an important scientific research area in the fields of hydrology, climate and geography [43]. Thus, monitoring the dynamic changes in water resources in this region has very important research value [44,45]. However, due to the particularity of climate and terrain, this region is often covered by clouds, and the clear-sky images are difficult to obtain in this region. Therefore, the study area is suitable to explore how cloud-shadow negative samples affect the accuracy of water body extraction, and is a challenging area to accurately extract water bodies.

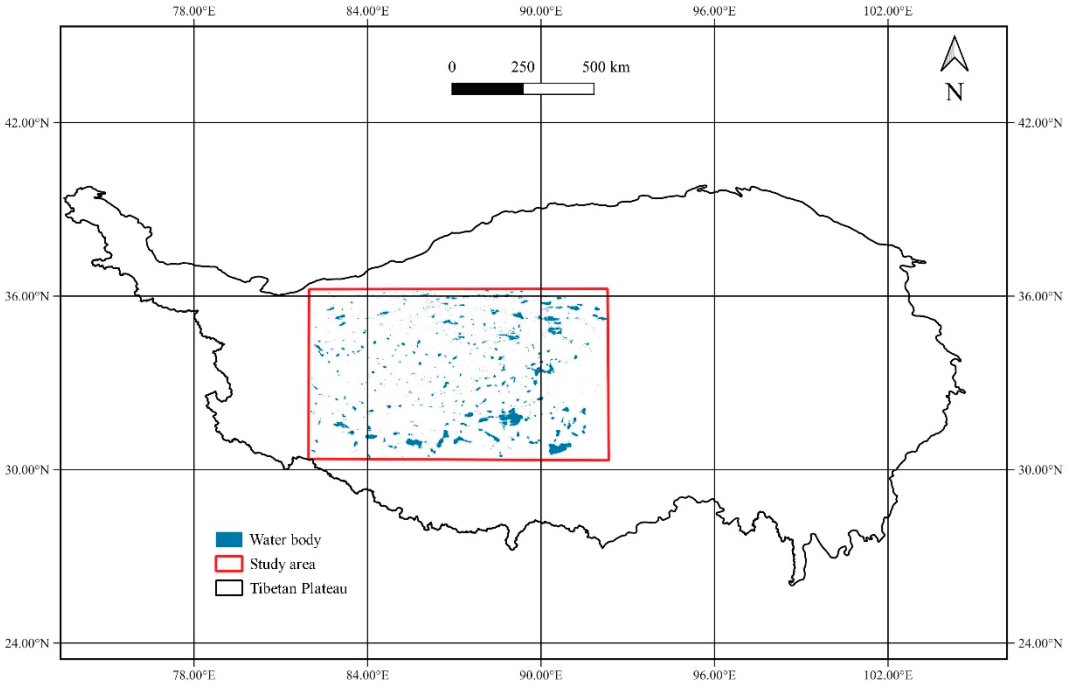


Figure 1. The location of the study area.

2.2. Data Sources

Sentinel-2 imagery is selected as the main data source in this study due to its high spatio-temporal resolution and image quality [46]. The Sentinel-2 image contains 13 spatial resolution bands of 10 m, 20 m and 60 m, which can thus satisfy the requirement of large-scale and accurate mapping. Its five-day revisiting period also greatly enhances the availability of the images. In this study, we choose three water-sensitive bands for deep learning networks to learn the features of water bodies [47]. They are Short Wave Infra-Red (SWIR), Near Infra-Red (NIR), and Red with 20 m, 10 m and 10 m resolution, respectively. The 20 m SWIR thus is upsampled to 10 m by using the nearest neighboring algorithm.

To accelerate the labeling work for water bodies, the European Space Agency (ESA) WorldCover 10 m 2020 product is used as the supplementary data [48]. The ESA WorldCover 10 m 2020 product is a global land cover product with a resolution of 10 m, which is the same as Sentinel-2 images in spatial resolution. This product contains 11 land cover categories, including permanent water bodies, and the permanent water bodies are extracted to assist in labeling water bodies based on the Sentinel-2 images.

2.3. Water Body Extraction Network

The structure of water body extraction network is shown in Figure 2. The Swin Transformer [49] network which fully implements the self-attention mechanism in the computer vision (CV) field, is used to learn the water body feature. The attention mechanism selectively imitates the human behavior of paying attention to information, focusing attention on important information, and globally connecting contextual information to understand it. Unlike CNN, which only focuses on local information, the network based on an attentional mechanism [50] can obtain different attention of water body in the image and extract water body features at multiple levels from global to local. Thus, the transformer-based network can better capture the overall connection and difference between water body and other ground objects [51].

The Swin Transformer network has four modules for automatically extracting feature maps. They are the Swin Transformer Block, Patch Partition module, Linear Embedding module, and Patch Merging module. The Swin Transformer Block includes Window Multihead Self-Attention (W-MSA) and Shifted-Window Multihead Self-Attention (SW-MSA) to calculate the global attention. The W-MSA focuses on calculating self-attention

within the windows, also known as local self-attention, and the SW-MSA can interact with the information by shifting to obtain the global self-attention [52]. The Patch Partition module is used to convert the input images into different patches and change the minimum unit of the image from pixel to patch. The function of the Linear Embedding module is to transform the image into a one-dimensional vector. The Patch Merging module serves to downscale the feature maps and is able to form pyramid feature maps. The pyramid feature maps include different levels of context features and thus are good for more accurately extracting water bodies [53].

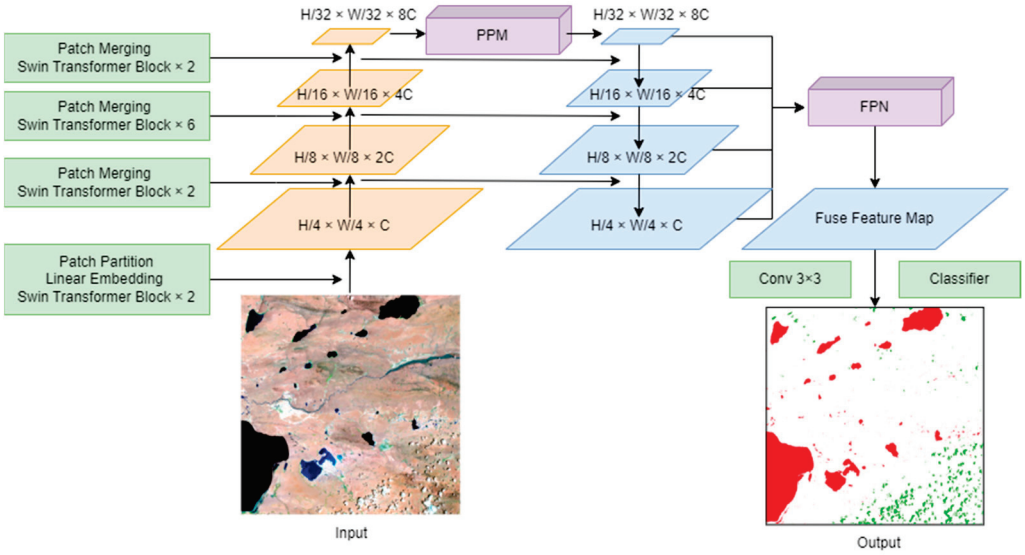


Figure 2. The structure of the water body extraction network.

To extract water bodies from the feature maps, Pyramid Pool Module (PPM) [54] is first used to obtain the scene-level feature, and Feature Pyramid Network (FPN) is used to combine the multi-scale object-level feature maps with the scene-level feature then obtain a fused feature map [55]. The fused feature map further goes through a 3×3 convolution, then a classifier is used to perform pixel-level classification.

2.4. Preparation for Training Data

The process of training data preparation is shown in Figure 3, which includes obtaining the SWIR-NIR-Red combined image data (SNR image data) from the Sentinel-2 imagery and obtaining ground truth data. The SNR image data is obtained by stacking the SWIR, NIR, Red bands from Sentinel-2 images.

To explore whether cloud shadow labels can improve the accuracy of water body extraction, two groups of training datasets are prepared: the group in which only water bodies are labeled, noted as the Water group, and the group in which both water bodies and cloud shadows are labeled, noted as the Water_Shadow group. The Water group includes the SNR image data and ground truth data that only water bodies are labeled, and the Water_Shadow group includes the SNR image data and ground truth data that both water bodies and cloud shadows are labeled.

To accelerate the labeling of water bodies, the permanent water bodies are extracted from the ESA World Cover 10 m 2020 product, then the water bodies are manually modified based on a selected Sentinel-2 image at a specific time. The modification includes the water boundary correction and addition of the missing water bodies.

During labeling the cloud shadows, no extra data, including the cloud or cloud shadow masks, was used as reference data. The cloud shadows are fully labeled manually based on the true color images from Sentinel-2 for ensuring the accuracy of cloud shadows.

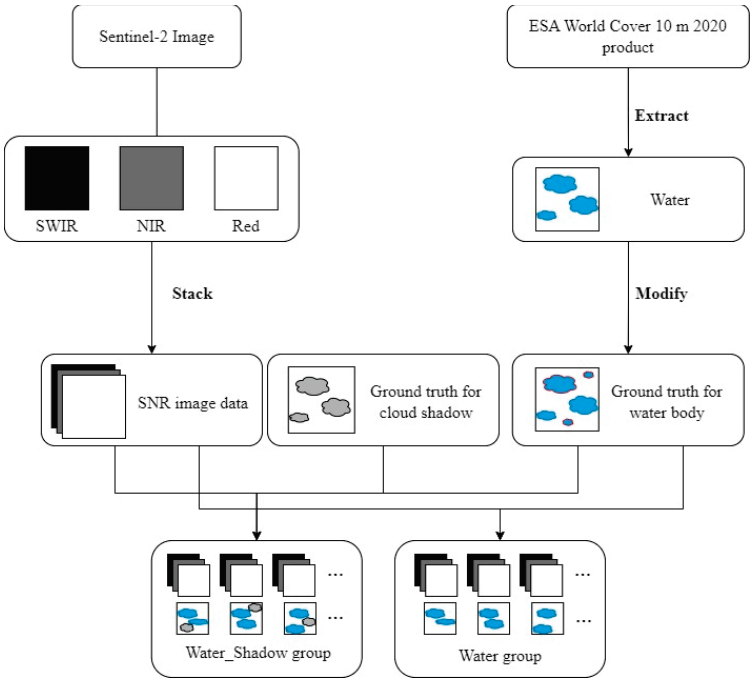


Figure 3. The process of preparation for training data.

3. Experiment and Results

3.1. Experiment Settings

Five Sentinel-2 images were selected to prepare for the training data, and the acquisition time of the five images is in the morning local time. A total of 8000 training samples were produced based on the Sentinel-2 images. The training samples are mainly located in the middle of the western Tibetan Plateau. Each of them is 272-pixel width and 272-pixel height due to the limitation of our GPU memory size. To explore how cloud shadows affect the accuracy of water body extraction, two training data groups are set up. One is the group that only water bodies are labeled, and the other is the group that both water bodies and cloud shadows are labeled. Each group includes two training datasets with 1% and 3% proportions of cloud shadows, respectively, as shown in Table 1. The proportion of water bodies in every training dataset is approximately 11% and the number of samples in all training datasets is 6400, which ensures the accuracy of water body extraction is not affected by the number of training samples and the proportion of water bodies.

Table 1. Training dataset settings.

Training Data Group	Training Dataset	Water Body		Cloud Shadow	
		Proportion	Labeled	Proportion	Labeled
Water group	Water_P1	11%	Yes	1%	No
	Water_P3	11%	Yes	3%	No
Water_Shadow group	Water_Shadow_P1	11%	Yes	1%	Yes
	Water_Shadow_P3	11%	Yes	3%	Yes

The validation dataset is made based on two additional Sentinel-2 images. The validation sets are also divided into a Water validation dataset and a Water_Shadow validation dataset. Each validation set contains 1012 samples with a size of 272 × 272 pixels, and the distribution of water bodies and cloud shadows are same. The only difference between the two validation datasets is that the cloud shadows are labeled in the Water_Shadow validation dataset.

The water body extraction network is trained for 230 epochs, and the top-three accuracy on the validation data are used to compare and analyze the performance of the network trained by different training data. The hardware environment in this study is an Intel(R) Xeon(R) W-2245 CPU @ 3.90 GHz with 16.0 GB RAM and an NVIDIA GeForce RTX3080 Ti GPU with 7424 CUDA cores. The initial learning rate is 0.000003, and it becomes 0.00006 by linearly changing small multiplicative factor after 10 epochs.

3.2. Metrics of Assessment

The metrics of assessment used in the experiment include the *Overall Accuracy* (*OA*), *mean Intersection Over Union* (*mIoU*), and Kappa.

The calculation formula of *OA* can be represented as:

$$OA = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

OA is calculated by the confusion matrix, where *FP* denotes false-positive pixels, *FN* denotes false-negative pixels, *TP* denotes true-positive pixels, *TN* denotes true-negative pixels, and *N* denotes the number of pixels.

The calculation formula of the *mIoU* is:

$$mIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{TP}{TP + FN + FP} \tag{2}$$

mIoU can also be calculated by a confusion matrix, where *k* is the number of classes.

The calculation formula of *Kappa* is:

$$Kappa = \frac{Po - Pe}{1 - Pe} \tag{3}$$

where *Po* is the *OA*, and *Pe* can be represented as:

$$Pe = \frac{a1 \times b1 + a2 \times b2 + \ldots + ax \times bx}{N \times N} \tag{4}$$

where *a1*, *a2*... *ax* are the numbers of true values for every class and *b1*, *b2*..., *bx* are the numbers of prediction values for every class, and the *N* is the number of pixels.

3.3. Results

3.3.1. Accuracy Evaluation

Table 2 shows the top-three accuracy evaluation results of the water body extraction network trained by the four training datasets, and the average values are obtained by calculating the values based on the top-three accuracy evaluation results.

Table 2. Results of accuracy evaluation.

	Training Dataset	OA	mIoU	Kappa	Epoch
Top 1	Water _P1	0.9970	0.9734	0.9728	180
	Water _P3	0.9976	0.9778	0.9774	138
	Water_Shadow _P1	0.9891	0.9185	0.9459	61
	Water_Shadow _P3	0.9917	0.9429	0.9589	69
Top 2	Water _P1	0.9962	0.9664	0.9654	199
	Water _P3	0.9974	0.9766	0.9761	191
	Water_Shadow _P1	0.9886	0.9081	0.9429	60
	Water_Shadow _P3	0.9899	0.9317	0.9507	92

Table 2. *Cont.*

	Training Dataset	OA	mIoU	Kappa	Epoch
Top 3	Water _P1	0.9940	0.9488	0.9464	211
	Water _P3	0.9968	0.9714	0.9707	135
	Water_Shadow _P1	0.9876	0.9035	0.9383	90
	Water_Shadow _P3	0.9892	0.9252	0.9472	70
Average	Water _P1	0.9957	0.9629	0.9615	-
	Water _P3	0.9973	0.9753	0.9747	-
	Water_Shadow _P1	0.9884	0.91	0.9424	-
	Water_Shadow _P3	0.9903	0.9333	0.9523	-

In general, the accuracy on all four training datasets is high. The average OA values on the four training datasets are 0.9957, 0.9973, 0.9884 and 0.9903. The average mIoU values are 0.9629, 0.9753, 0.91 and 0.9333, respectively. The average Kappa values are 0.9615, 0.9747, 0.9424 and 0.9523, respectively. Thus, the overall performance of water body extraction using our water body extraction network is satisfactory. The training datasets in which only water bodies are labeled have higher values than the training datasets in which both water bodies and cloud shadows are labeled, which indicates that labeling cloud shadows will decrease the values of the metrics. When cloud shadows are not labeled, the "Water_P3" dataset has higher OA, mIoU and Kappa values than the "Water_P1" dataset in all results. When cloud shadows are labeled, the "Water_Shadow_P3" dataset has higher OA, mIoU and Kappa values than the "Water_Shadow_P1" dataset in all results. Thus, when the proportions of water body are similar, increasing the proportion of cloud shadows can improve the accuracy of the network. From the results, Water_P3 has the highest values. When cloud shadows are not labeled and the proportion is 3%, the network has the best accuracy.

For the epochs of training, the highest accuracy results in the four experimental sets appear in epochs 180, 199 and 211, epochs 138, 135 and 191, epochs 61, 60 and 90, and epochs 69, 92 and 70. This result shows that when cloud shadows are labeled, the network will converge faster.

3.3.2. Prediction Results

Five cases are summarized to illustrate the prediction results.

(1) Cloud shadows being misclassified as water bodies

Results for the Water Group: when cloud shadows are not labeled, the water body extraction network trained by the "Water_P1" dataset will be disturbed by cloud shadows. Some cloud shadows are misclassified as water bodies, especially the part in the middle of the cloud shadows with a large area, as shown in Figure 4b. However, when the proportion of unlabeled cloud shadows is 3%, the phenomenon of cloud shadows being misclassified as water bodies is significantly reduced, and the interference of cloud shadows is well eliminated (Figure 4c).

Results for the Water_Shadow Group: when cloud shadows are labeled, the water body extraction network can predict most boundaries of cloud shadows, as shown in Figure 4d. The phenomenon of cloud shadows being misclassified as water bodies is less when the proportion is 3% than when the proportion is 1% (Figure 4e). However, compared with experimental sets in which cloud shadows are not labeled, the areas in the middle of large cloud shadows are still misclassified as water bodies. Therefore, labeling cloud shadows does not make cloud shadows less misclassified than adding cloud shadows but without labeling.

(2) Precision of water boundaries

Results for the Water Group: when cloud shadows are not labeled and the proportion is 1%, there are some noise points on the boundary of lakes, as shown in Figure 5b. However, when the proportion of cloud shadows increases to 3%, the noise points on the boundary are eliminated, and the extracted lake boundaries are more accurate (Figure 5c). Thus, increasing the proportion of cloud shadows in the training dataset helps to extract more accurate water boundaries.

Results for the Water_Shadow Group: when cloud shadows are labeled, regardless of whether the proportion of cloud shadows is 1% or 3%, the results are identical to those for the Water_P3 dataset, as shown in Figure 5d,e. The water boundaries are also accurately predicted. Thus, when cloud shadows are labeled, the water boundaries are more accurately predicted, and the proportion does not affect the accuracy of the boundaries.

(3) Water bodies being missed during extraction

Results for the Water Group: when cloud shadows are not labeled, the water body extraction network trained by 1% cloud shadows and 3% cloud shadows in the training dataset can predict most water bodies, as shown in Figure 6b,c, even the small lakes that cover several pixels. This result indicates that when cloud shadows are not labeled, the proportion of cloud shadows hardly affects the number of predicted water bodies.

Results for the Water_Shadow Group: when cloud shadows are labeled, the number of predicted water bodies decreases compared with those of the Water group, as shown in Figure 6d,e. Thus, labeling cloud shadows reduces the number of predicted water bodies.

When the proportion of labeled cloud shadows is 3%, the number of predicted water bodies by the network is less than that predicted by the network when the proportion of cloud shadows is 1%. Thus, when cloud shadows are labeled, the proportion of cloud shadows in the training dataset is higher, and the number of predicted water bodies is fewer.

(4) The middle part of islands being misclassified as water body

Results for the Water Group: when cloud shadows are not labeled and the proportion is 1%, the middle part of the islands in the lakes is misclassified as water body, as shown in Figure 7b. However, this phenomenon disappears when the proportion of cloud shadows increases to 3% (Figure 7c). Thus, when cloud shadows are not labeled, increasing the proportion of cloud shadows in the training dataset can make the water body extraction network better predict the islands.

Results for the Water_Shadow Group: when cloud shadows are labeled, regardless of whether the proportion of cloud shadows is 1% or 3%, the prediction result of the islands is identical to that of the Water_P3 dataset, as shown in Figure 7d,e. The middle part of islands is not misclassified as water body, and the prediction result is more accurate. Thus, labeling cloud shadows can make the water body extraction network better distinguish the islands from the water bodies.

(5) Cloud shadow extraction results

When comparing our extraction results of cloud shadows with those of Fmask, the boundaries of our cloud shadow results are more refined than the boundaries extracted by Fmask. The cloud shadow boundaries extracted by Fmask are coarser than the true boundaries, as shown in Figure 8b.

When the proportion of cloud shadows is 1% in the training dataset, our deep learning network can predict most cloud shadow boundaries. However, the extracted cloud shadows are mostly fragmented and incomplete, as shown in Figure 8c. When the proportion of cloud shadows is increased to 3%, the boundaries of cloud shadows are more accurate and complete than when the proportion is 1%, as shown in Figure 8d. Thus, increasing the proportion of cloud shadows can improve the accuracy and completeness of extracted cloud shadows.

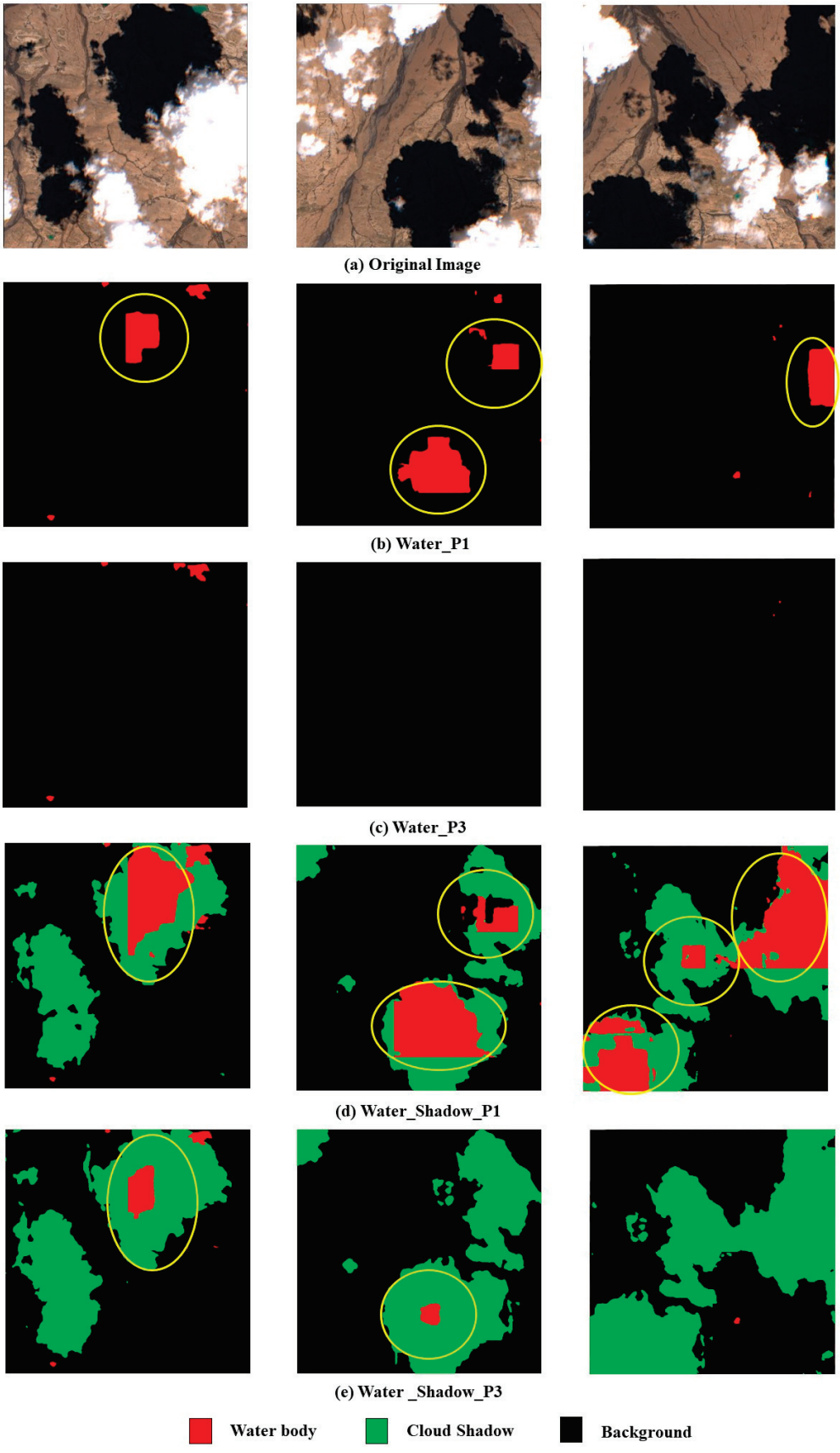


Figure 4. Results of cloud shadows being misclassified as water bodies (The areas circled in yellow represent the cloud shadows being misclassified as water bodies).

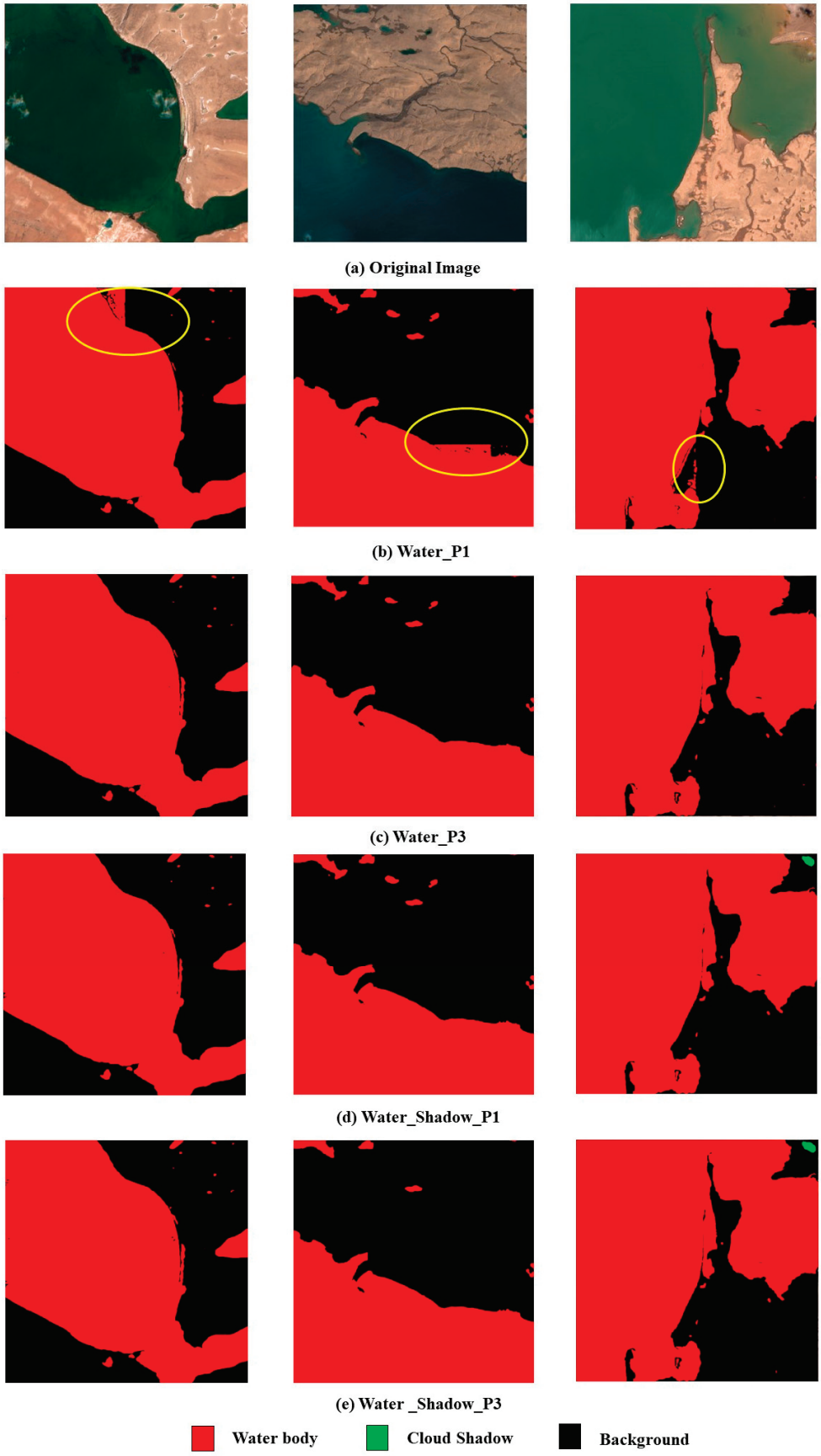


Figure 5. Results of water boundary extraction (The areas circled in yellow represent the noise points on the boundaries of water bodies).

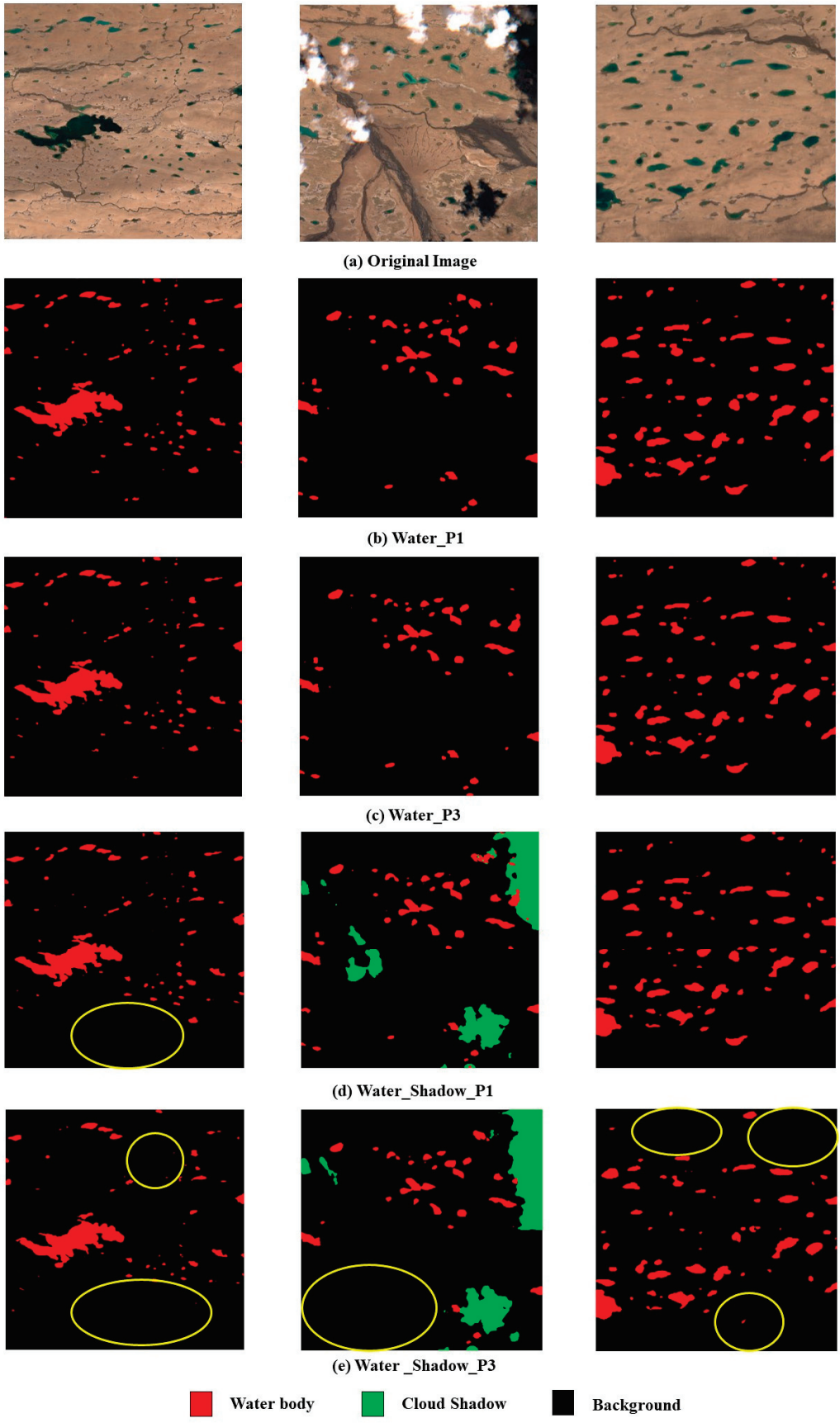


Figure 6. Results of water bodies being missed (The areas circled in yellow represent the water bodies being missed).

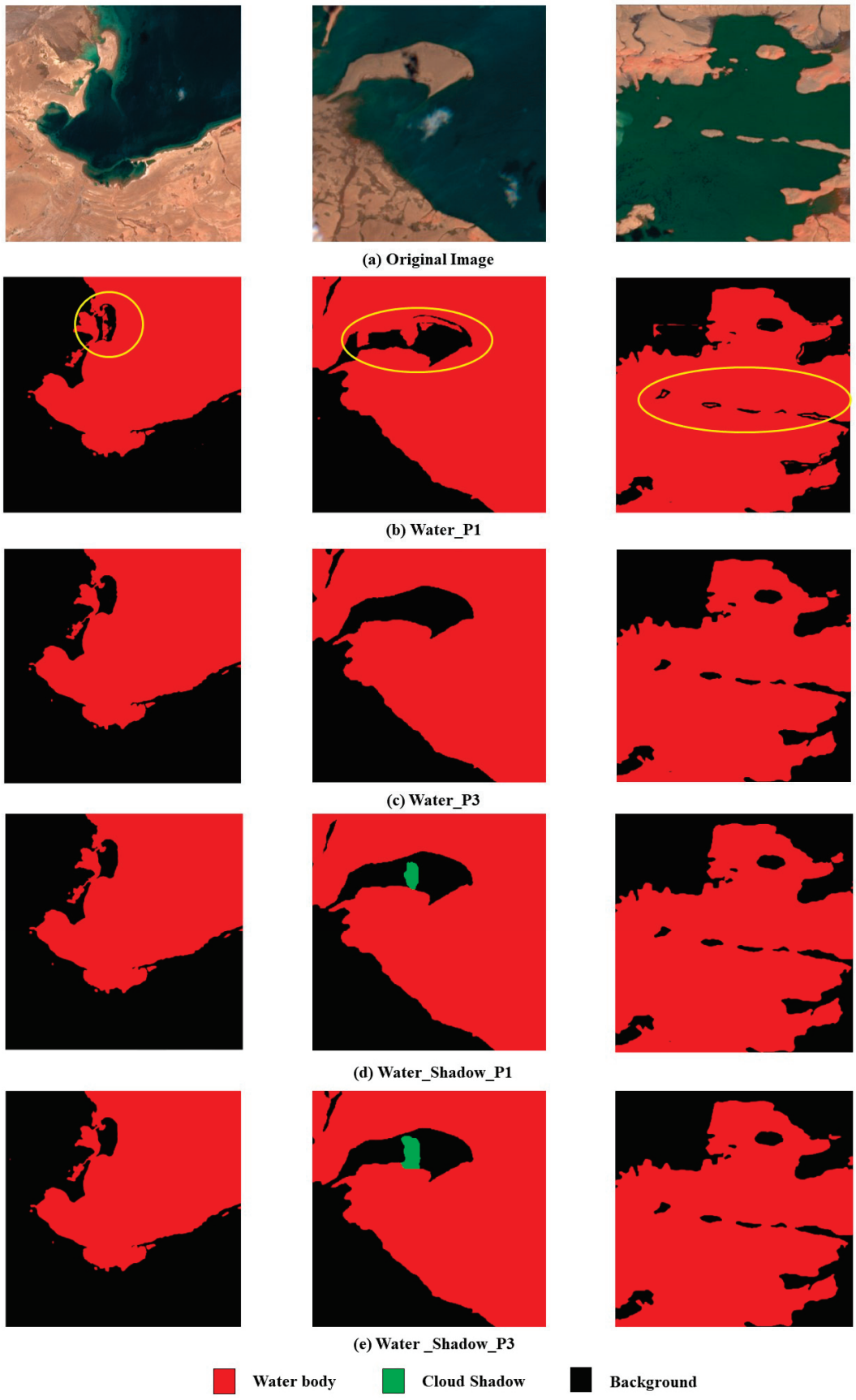


Figure 7. Results of the middle part of islands being misclassified as water body (The areas circled in yellow represent the middle part of islands being misclassified as water body).

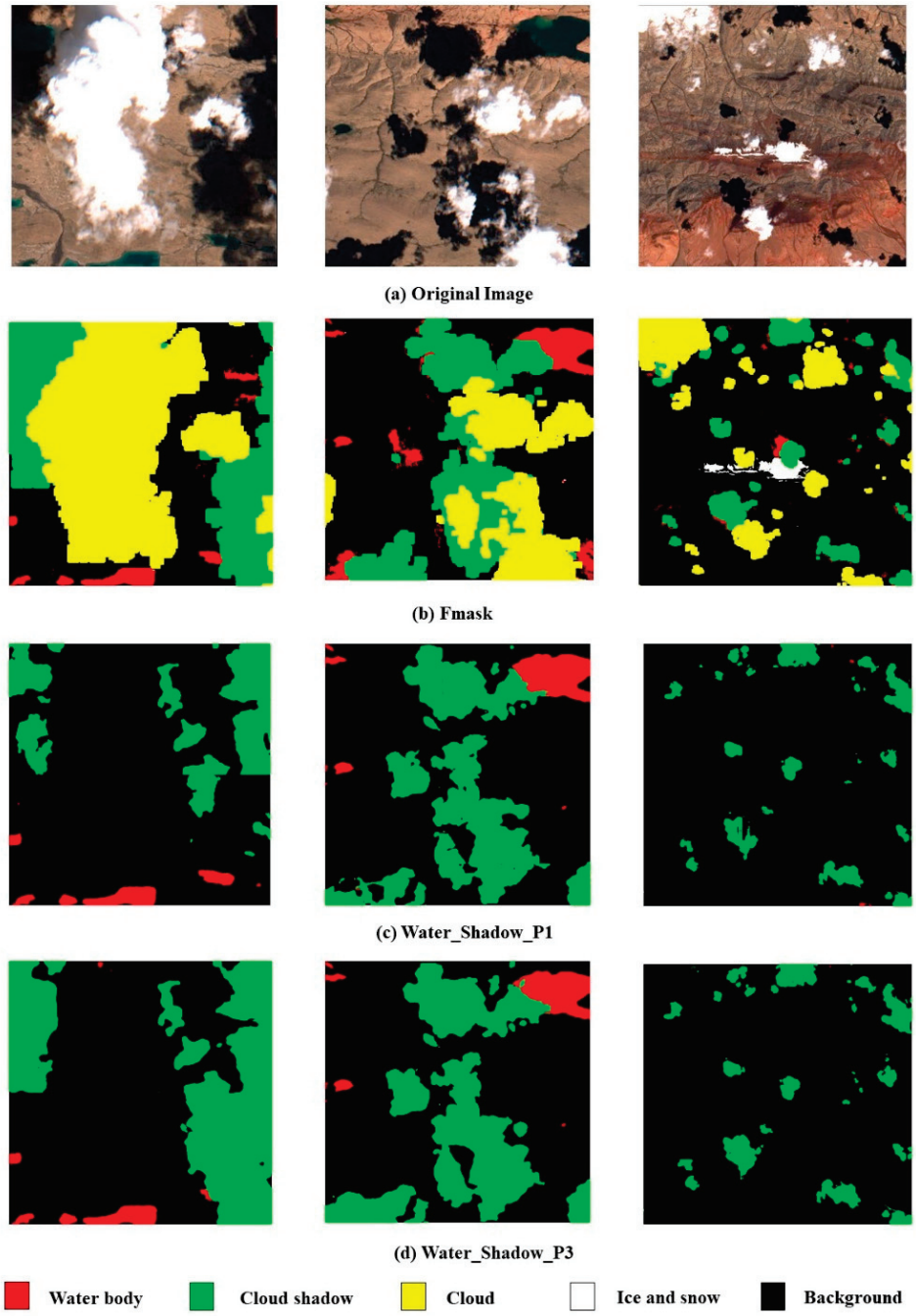


Figure 8. Results of cloud shadow extraction.

4. Discussion

While the existing studies pay more attention to the improvement or innovation of deep learning models for accurate water body extraction, this study highlights the function of negative samples. It reveals that the negative samples play an import role in achieving good or satisfactory accuracy, and confirms that cloud shadow negative samples do affect the accuracy of water body extraction from remote sensing images. More specifically, the study demonstrates an effective method based on the negative samples to address the misclassification of cloud shadows thereby improving the accuracy of water body extraction. Based on the experiment of this study, it can be seen that even if the negative samples change slightly (the proportion increases from 1% to 3% in this study), the prediction result of water bodies changes significantly. A more detailed discussion about the cloud shadow negative samples being categorized into cloud shadow misclassification, the precision of

the water boundaries, small water body extraction, as well as the precision of the extracted cloud shadows follows:

Cloud shadow misclassification: the cloud shadow negative samples help to address the issue of cloud shadow misclassification in water body extraction. When 1% cloud shadow negative samples are introduced, fewer cloud shadows are misclassified as water bodies despite that there are still a few cloud shadows misclassified. When the proportion of the cloud shadows reaches 3% and even if the cloud shadows are not labeled, the water body extraction network is able to handle the misclassification of cloud shadows. In a word, a higher proportion of cloud shadow negative samples in the training datasets corresponds to a stronger ability of the network to eliminate the interference of cloud shadows. In addition, labeling the cloud shadows cannot significantly further reduce the cloud shadow misclassification. We believe the reason is that prediction of both water bodies and cloud shadows is more challenging than only predicting water bodies for the network. Thus, the accuracy of water body extraction could be even slightly low when extracting both water bodies and cloud shadows at the same time.

Precision of the water boundaries: the experiment shows that when the proportion of cloud shadows is 1%, labeling cloud shadows helps to extract more accurate water boundaries, and when the proportion of cloud shadows increases to 3%, both labeling and not labeling cloud shadows can further improve the precision of the water boundaries. We believe that when the cloud shadow negative samples in the training samples reach a certain proportion (3% in our experiment), the water body extraction network learned more features of cloud shadows and thus can accurately distinguish water bodies from cloud shadows thereby extracting more accurate water boundaries.

Small water body extraction: the experiments show that labeling the cloud shadows causes the small water bodies to be missed in water body extraction. The feature of small water bodies is obscure relative to large water bodies, and they are more easily missed when the network conducts more challenging tasks such as multi-classification. Therefore, if the proportion of cloud shadows is sufficiently high, it is more conducive to water body extraction without labeling cloud shadows.

Precision of the extracted cloud shadows: with the training datasets in which cloud shadows are labeled, the experiment proved that the boundary of cloud shadows identified by the proposed transformer-based network is more accurate and refined than the ones identified by Fmask [40]. Although, the proportion of cloud shadows in the training dataset is only 3% and not high, the network can extract both water bodies and cloud shadows at the same time. This demonstrates the great potential of the novel transformer-based deep learning networks applying in remote sensing field.

Lastly, although the network achieves the best performance on water body extraction with 3% cloud shadow negative samples, the proportion of 3% is only applicable to cloud shadow samples, and meets 6400 samples and every image size of 272 $\times$ 272 pixels. We believe that the method of this study could be applied on other negative samples such as mountain shadows and building shadows for eliminating the interference of them in water body extraction. When applying other negative samples, the 3% may not be an appropriate proportion of negative samples. However, the appropriate proportion of other negative samples can be explored based on the method of this study.

5. Conclusions

Deep learning is one of the most effective approaches to extract water bodies from remote sensing images, and making samples for training the deep learning networks is indispensable. This study pays close attention to the negative samples to explore the impact of the cloud shadow negative samples on the accuracy of water body extraction. The western Tibetan Plateau is the study area for producing training data and validating prediction results. A water body extraction based on a novel vision transformer network was built, and the network was trained with the datasets that contain different proportions of cloud shadows. It was found that the training datasets containing a certain proportion

of cloud shadows could help the water body extraction network to better distinguish water bodies and cloud shadows, and make the extracted water bodies more accurate. With 3% cloud shadow negative samples in this study, the water body extraction network can well address the issue of cloud shadows being misclassified as the water bodies, and the evaluation results over the validation dataset achieve *OA* of 0.9973, *mIoU* of 0.9753, and *Kappa* of 0.9747. The network was also trained with the datasets in which cloud shadows were labeled to investigate whether labeling the cloud shadows is helpful. It was found that labeling the cloud shadows in the training data is unnecessary. In addition, this study also reveals the cloud shadow boundaries predicted by our network are more accurate than the ones using the Fmask method. In the future, we will conduct more studies on other negative samples, such as mountain shadows and building shadows, to investigate whether they have the similar results to this study, and what the appropriate proportion of them is. In addition, we may develop other metrics to quantify the spectral and morphological features of negative samples and better explain the function of negative samples in improving the performance of deep learning networks.

Author Contributions: J.S.: conceptualization, methodology, software, resources, funding acquisition, writing—review and editing, supervision. X.Y.: methodology, validation, data curation, formal analysis, visualization, writing—original draft. All authors have read and agreed to the published version of the manuscript.

Funding: The National Key Research and Development Program of China (2022YFF0711602, 2021YFE0117800), the 14th Five-year Informatization Plan of Chinese Academy of Sciences (CAS-WX2021SF-0106), and the National Data Sharing Infrastructure of Earth System Science (http://www.geodata.cn/ accessed on 16 July 2022).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data generated and analyzed during this study are available from the corresponding author by request.

Acknowledgments: We appreciate the detailed comments from the editor and the anonymous reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, Z.; Prinet, V.; Ma, S. Water Body Extraction from Multi-Source Satellite Images. In Proceedings of the IGARSS 2003—2003 IEEE International Geoscience and Remote Sensing Symposium, Toulouse, France, 21–25 July 2003; Volume 6, pp. 3970–3972. [CrossRef]
2. Li, J.; Ma, R.; Cao, Z.; Xue, K.; Xiong, J.; Hu, M.; Feng, X. Satellite Detection of Surface Water Extent: A Review of Methodology. *Water* **2022**, *14*, 1148. [CrossRef]
3. Shrestha, R.; Di, L. Land/Water Detection and delineation with Landsat Data Using Matlab/ENVI. In Proceedings of the 2013 Second International Conference on Agro-Geoinformatics, Fairfax, VA, USA, 12–16 August 2013; pp. 211–214. [CrossRef]
4. Sit, M.A.; Demiray, B.Z.; Xiang, Z.; Ewing, G.J.; Sermet, Y.; Demir, I. A comprehensive review of deep learning applications in hydrology and water resources. *Water Sci. Technol.* **2020**, *82*, 2635–2670. [CrossRef] [PubMed]
5. Li, M.; Hong, L.; Guo, J.; Zhu, A. Automated Extraction of Lake Water Bodies in Complex Geographical Environments by Fusing Sentinel-1/2 Data. *Water* **2022**, *14*, 30. [CrossRef]
6. Li, J.; Meng, Y.; Li, Y.; Cui, Q.; Yang, X.; Tao, C.; Wang, Z.; Li, L.; Zhang, W. Accurate water extraction using remote sensing imagery based on normalized difference water index and unsupervised deep learning. *J. Hydrol.* **2022**, *612*, 128202. [CrossRef]
7. Shen, C. A Transdisciplinary Review of Deep Learning Research and Its Relevance for Water Resources Scientists. *Water Resour. Res.* **2018**, *54*, 8558–8593. [CrossRef]
8. Yu, L.; Zhang, R.; Tian, S.; Yang, L.; Lv, Y. Deep Multi-Feature Learning for Water Body Extraction from Landsat Imagery. *Autom. Control. Comput. Sci.* **2018**, *52*, 517–527. [CrossRef]
9. Wang, Z.; Gao, X.; Zhang, Y.; Zhao, G. MSLWENet: A Novel Deep Learning Network for Lake Water Body Extraction of Google Remote Sensing Images. *Remote Sens.* **2020**, *12*, 4140. [CrossRef]
10. Yuan, K.; Zhuang, X.; Schaefer, G.; Feng, J.; Guan, L.; Fang, H. Deep-Learning-Based Multispectral Satellite Image Segmentation for Water Body Detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 7422–7434. [CrossRef]

11. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, Honolulu, HI, USA, 21–26 July 2017; Volume 39, pp. 640–651.
12. Li, L.; Yan, Z.; Shen, Q.; Cheng, G.; Gao, L.; Zhang, B. Water Body Extraction from Very High Spatial Resolution Remote Sensing Data Based on Fully Convolutional Networks. *Remote Sens.* **2019**, *11*, 1162. [CrossRef]
13. Zhang, J.; Xing, M.; Sun, G.-C.; Chen, J.; Li, M.; Hu, Y.; Bao, Z. Water Body Detection in High-Resolution SAR Images With Cascaded Fully-Convolutional Network and Variable Focal Loss. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *59*, 316–332. [CrossRef]
14. Song, A.; Kim, Y.; Kim, Y. Change detection of surface water in remote sensing images based on fully convolutional network. *J. Coast. Res.* **2019**, *91*, 426–430. [CrossRef]
15. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Lecture Notes in Computer Science, Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015*; Springer: Cham, Switzerland, 2015; Volume 9351. [CrossRef]
16. Ch, A.; Ch, R.; Gadamsetty, S.; Iwendi, C.; Gadekallu, T.R.; Dhaou, I.B. ECDSA-Based Water Bodies Prediction from Satellite Images with UNet. *Water* **2022**, *14*, 2234. [CrossRef]
17. An, S.; Rui, X. A High-Precision Water Body Extraction Method Based on Improved Lightweight U-Net. *Remote Sens.* **2022**, *14*, 4127. [CrossRef]
18. Jiang, C.; Zhang, H.; Wang, C.; Ge, J.; Wu, F. Water Surface Mapping from Sentinel-1 Imagery Based on Attention-UNet3+: A Case Study of Poyang Lake Region. *Remote Sens.* **2022**, *14*, 4708. [CrossRef]
19. Ge, C.; Xie, W.; Meng, L. Extracting Lakes and Reservoirs from GF-1 Satellite Imagery Over China Using Improved U-Net. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 1504105. [CrossRef]
20. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
21. James, T.; Schillaci, C.; Lipani, A. Convolutional neural networks for water segmentation using sentinel-2 red, green, blue (RGB) composites and derived spectral indices. *Int. J. Remote Sens.* **2021**, *42*, 5338–5365. [CrossRef]
22. Wu, H.; Song, H.; Huang, J.; Zhong, H.; Zhan, R.; Teng, X.; Qiu, Z.; He, M.; Cao, J. Flood Detection in Dual-Polarization SAR Images Based on Multi-Scale Deeplab Model. *Remote Sens.* **2022**, *14*, 5181. [CrossRef]
23. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
24. Balaska, V.; Bampis, L.; Kansizoglou, I.; Gasteratos, A. Enhancing satellite semantic maps with ground-level imagery. *Robot. Auton. Syst.* **2021**, *139*, 103760. [CrossRef]
25. Weng, L.; Xu, Y.; Xia, M.; Zhang, Y.; Liu, J.; Xu, Y. Water Areas Segmentation from Remote Sensing Images Using a Separable Residual SegNet Network. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 256. [CrossRef]
26. Pu, F.; Ding, C.; Chao, Z.; Yu, Y.; Xu, X. Water-Quality Classification of Inland Lakes Using Landsat8 Images by Convolutional Neural Networks. *Remote Sens.* **2019**, *11*, 1674. [CrossRef]
27. Wang, Y.; Li, Z.; Zeng, C.; Xia, G.-S.; Shen, H. An Urban Water Extraction Method Combining Deep Learning and Google Earth Engine. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 769–782. [CrossRef]
28. Guo, H.; He, G.; Jiang, W.; Yin, R.; Yan, L.; Leng, W. A Multi-Scale Water Extraction Convolutional Neural Network (MWEN) Method for GaoFen-1 Remote Sensing Images. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 189. [CrossRef]
29. Luo, X.; Tong, X.; Hu, Z. An applicable and automatic method for earth surface water mapping based on multispectral images. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *103*, 102472. [CrossRef]
30. Fei, J.; Liu, J.; Ke, L.; Wang, W.; Wu, P.; Zhou, Y. A deep learning-based method for mapping alpine intermittent rivers and ephemeral streams of the Tibetan Plateau from Sentinel-1 time series and DEMs. *Remote Sens. Environ.* **2022**, *282*, 113271. [CrossRef]
31. Li, M.; Wu, P.; Wang, B.; Park, H.; Hui, Y.; Yanlan, W. A Deep Learning Method of Water Body Extraction From High Resolution Remote Sensing Images with Multisensors. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 3120–3132. [CrossRef]
32. Chen, F. Comparing Methods for Segmenting Supra-Glacial Lakes and Surface Features in the Mount Everest Region of the Himalayas Using Chinese GaoFen-3 SAR Images. *Remote Sens.* **2021**, *13*, 2429. [CrossRef]
33. Yang, F.; Guo, J.; Tan, H.; Wang, J. Automated Extraction of Urban Water Bodies from ZY-3 Multi-Spectral Imagery. *Water* **2017**, *9*, 144. [CrossRef]
34. Dirscherl, M.; Dietz, A.J.; Kneisel, C.; Kuenzer, C. A Novel Method for Automated Supraglacial Lake Mapping in Antarctica Using Sentinel-1 SAR Imagery and Deep Learning. *Remote Sens.* **2021**, *13*, 197. [CrossRef]
35. Jiang, D.; Li, X.; Zhang, K.; Marinsek, S.; Hong, W.; Wu, Y. Automatic Supraglacial Lake Extraction in Greenland Using Sentinel-1 SAR Images and Attention-Based U-Net. *Remote Sens.* **2022**, *14*, 4998. [CrossRef]
36. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
37. Kansizoglou, I.; Bampis, L.; Gasteratos, A. Deep feature space: A geometrical perspective. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 6823–6838. [CrossRef] [PubMed]
38. Wang, G.; Wu, M.; Wei, X.; Song, H. Water Identification from High-Resolution Remote Sensing Images Based on Multidimensional Densely Connected Convolutional Neural Networks. *Remote Sens.* **2020**, *12*, 795. [CrossRef]

39. Feyisa, G.L.; Meilby, H.; Fensholt, R.; Proud, S.R. Automated Water Extraction Index: A new technique for surface water mapping using Landsat imagery. *Remote Sens. Environ.* **2014**, *140*, 23–35. [CrossRef]
40. Zhu, Z.; Woodcock, C.E. Object-based cloud and cloud shadow detection in Landsat imagery. *Remote Sens. Environ.* **2012**, *118*, 83–94. [CrossRef]
41. Zhu, Z.; Woodcock, C.E. Automated cloud, cloud shadow, and snow detection in multitemporal Landsat data: An algorithm designed specifically for monitoring land cover change. *Remote Sens. Environ.* **2014**, *152*, 217–234. [CrossRef]
42. Kang, S.; Xu, Y.; You, Q.; Flügel, W.-A.; Pepin, N.; Yao, T. Review of climate and cryospheric change in the Tibetan Plateau. *Environ. Res. Lett.* **2010**, *5*, 015101. [CrossRef]
43. Zhang, G.; Xie, H.; Kang, S.; Yi, D.; Ackley, S.F. Monitoring lake level changes on the Tibetan Plateau using ICESat altimetry data (2003–2009). *Remote Sens. Environ.* **2011**, *115*, 1733–1742. [CrossRef]
44. Song, C.; Huang, B.; Ke, L.; Richards, K.S. Remote sensing of alpine lake water environment changes on the Tibetan Plateau and surroundings: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 26–37. [CrossRef]
45. Zhang, G.; Li, J.; Zheng, G. Lake-area mapping in the Tibetan Plateau: An evaluation of data and methods. *Int. J. Remote Sens.* **2017**, *38*, 742–772. [CrossRef]
46. Drusch, M.; Del Bello, U.; Carlier, S.; Colin, O.; Fernandez, V.; Gascon, F.; Hoersch, B.; Isola, C.; Laberinti, P.; Martimort, P.; et al. Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. *Remote Sens. Environ.* **2012**, *120*, 25–36. [CrossRef]
47. Zhang, M.; Wang, X.; Shi, C.; Yan, D. Automated Glacier Extraction Index by Optimization of Red/SWIR and NIR /SWIR Ratio Index for Glacier Mapping Using Landsat Imagery. *Water* **2019**, *11*, 1223. [CrossRef]
48. Zanaga, D.; van de Kerchove, R.; de Keersmaecker, W.S.N.; Brockmann, C.; Quast, R.; Wevers, J.; Grosu, A.; Paccini, A.; Vergnaud, S.; Cartus, O.; et al. *ESA WorldCover 10 m 2020 v100*; The European Space Agency: Paris, France, 2021. [CrossRef]
49. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 9992–10002. [CrossRef]
50. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; Volume 30. [CrossRef]
51. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
52. Vaswani, A.; Ramachandran, P.; Srinivas, A.; Parmar, N.; Hechtman, B.; Shlens, J. Scaling local self-attention for parameter efficient visual backbones. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; p. 12889. [CrossRef]
53. He, X.; Zhou, Y.; Zhao, J.; Zhang, D.; Yao, R.; Xue, Y. Swin Transformer Embedding UNet for Remote Sensing Image Semantic Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4408715. [CrossRef]
54. Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; Sun, J. Unified Perceptual Parsing for Scene Understanding. In *Lecture Notes in Computer Science, Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018*; Springer International Publishing: Cham, Switzerland, 2018; pp. 432–448. [CrossRef]
55. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

remote sensing

MDPI

Article

Mean Inflection Point Distance: Artificial Intelligence Mapping Accuracy Evaluation Index—An Experimental Case Study of Building Extraction

Ding Yu [1], Aihua Li [1], Yinping Long [2], Yan Xu [3], Jinrui Li [4] and Xiongwu Xiao [3,*]

[1] Xi'an Research Institute of High Technology, Xi'an 710025, China; 200910483@stu.xust.edu.cn (D.Y.); aihuali_1@126.com (A.L.)
[2] College of Resources and Environment, Chengdu University of Information Technology, Chengdu 610225, China; longyp@cuit.edu.cn
[3] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, Wuhan 430072, China; xuyanlngd@whu.edu.cn
[4] Hubei Key Laboratory of Petroleum Geochemistry and Environment, Yangtze University, Wuhan 430100, China; 2021720585@yangtzeu.edu.cn
* Correspondence: xwxiao@whu.edu.cn

Abstract: Mapping is a fundamental application of remote sensing images, and the accurate evaluation of remote sensing image information extraction using artificial intelligence is critical. However, the existing evaluation method, based on Intersection over Union (IoU), is limited in evaluating the extracted information's boundary accuracy. It is insufficient for determining mapping accuracy. Furthermore, traditional remote sensing mapping methods struggle to match the inflection points encountered in artificial intelligence contour extraction. In order to address these issues, we propose the mean inflection point distance (MPD) as a new segmentation evaluation method. MPD can accurately calculate error values and solve the problem of multiple inflection points, which traditional remote sensing mapping cannot match. We tested three algorithms on the Vaihingen dataset: Mask R-CNN, Swin Transformer, and PointRend. The results show that MPD is highly sensitive to mapping accuracy, can calculate error values accurately, and is applicable for different scales of mapping accuracy while maintaining high visual consistency. This study helps to assess the accuracy of automatic mapping using remote sensing artificial intelligence.

Keywords: segmentation evaluation; deep learning; mean inflection point

1. Introduction

Remote sensing images with high resolution are widely used in many applications, including land cover mapping [1–3], urban classification analysis [4,5] and automatic road detection [6–9]. Semantic segmentation is an essential technology in computer vision, which assigns each pixel in an image to a different semantic category, such as buildings, roads, trees, and other objects. Moreover, it has significant application value in remote sensing image processing for building extraction. The performance of semantic segmentation is improving as deep learning technology advances. Many deep learning-based semantic segmentation algorithms, such as FCN [10] (Fully Convolutional Network), SegNet [11] (Segmentation Network), U-Net [12], U-Net++ [13], and DeepLab [14], among others, have emerged recently. These algorithms have achieved excellent results on various datasets and have even surpassed human performance on some datasets.

Instance segmentation [15] is an essential technology in computer vision that can simultaneously recognize and segment multiple objects in an image at the pixel level and then assign a unique identifier to each object. In building extraction [16–19], traditional methods first require semantic segmentation. Then, a further step is needed to refine the segmentation boundary to achieve building extraction [20,21]. On the other hand,

instance segmentation techniques can directly separate each building, avoiding boundary uncertainty and improving extraction accuracy and efficiency.

For example, researchers working on instance segmentation applications, which require an algorithm to delineate objects with pixel-level binary masks, have improved the standard average precision (AP) metric in COCO (Microsoft Common Objects in Context) [22] by 86% (relative) from 2015 to 2022. The current artificial intelligence mapping accuracy evaluation metrics generally use visual deep learning accuracy evaluation metrics: IoU, Generalized Intersection over Union (GIoU) [23], distance-IoU (DIoU) [24], Boundary IoU [25], and IoU-based instance segmentation metrics including AP (average accuracy) [26] and mAP [26], whereas mean surface distance (MSD) [27] is a commonly used evaluation metric in medical image segmentation. Accurate boundary information is a prerequisite for remote sensing mapping [28]. The widely used IoU and other improved methods evaluate only area coverage, but lack the ability to evaluate contour and inflection accuracy [29]. As shown in Figure 1, we wanted to evaluate the accuracy of the building segmentation contours extracted by deep learning by determining the two contours that need to be evaluated for accuracy. However, the current evaluation metrics have limited sensitivity to errors in object vector contour boundaries.

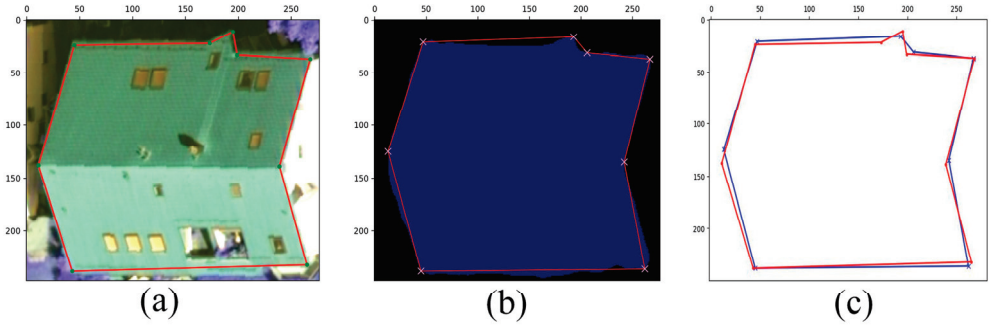


Figure 1. Vector image accuracy evaluation: (**a**) original image and reference contours; (**b**) deep learning extracted mask and regularized contours; (**c**) the two contours to be evaluated.

The IoU, also known as the Jaccard index, is the most commonly used evaluation metric in visual target detection benchmarks and is the most common metric for comparing the similarity between two arbitrary shapes, $A, B \subseteq \mathbb{R}^n$, which is achieved by the following equation:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{1}$$

For example, in the PASCAL VOC (Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes) [30] challenge, the widely reported detection accuracy metric, mean accuracy (mAP), is calculated based on a fixed IoU threshold (i.e., 0.5). However, the arbitrary choice of the IoU threshold does not adequately reflect the performance of different methods. Any localization accuracy above the threshold is treated equally. Thus, to reduce the sensitivity of this performance metric to the choice of the IoU threshold [25], the MS COCO benchmark test challenge average was mapped to multiple IoU thresholds.

The advantages of IoU include non-negativity, homogeneity, symmetry, and triangle inequality. In particular, IoU is invariant to scale, and therefore all the performance metrics used to evaluate segmentation [23], object detection [31–33], and tracking [34,35] rely on this metric.

However, IoU has a few disadvantages [25]: 1. It cannot measure whether the two comparison objects are adjacent or distant. 2. It cannot reflect the intersection mode of the two comparison objects. 3. It cannot reflect the distance relationship between the inflection point and the edge of the contour of the comparison object. 4. It can only perform an indirect evaluation of the accuracy of artificial intelligence mapping, and cannot calculate the error

distance and medium error value of the mapping. As shown in Figure 2, BoundaryIoU is equivalent to MSD when d is set to one pixel. Meanwhile, MSD and MPD are strongly associated. However, MSD is only suitable for raster images, and some points cannot be matched to the shortest distance point during calculation due to a lack of a matching contour. To summarize, IoU, BoundaryIoU, and MSD have limitations and are unsuitable for evaluating AI mapping accuracy.

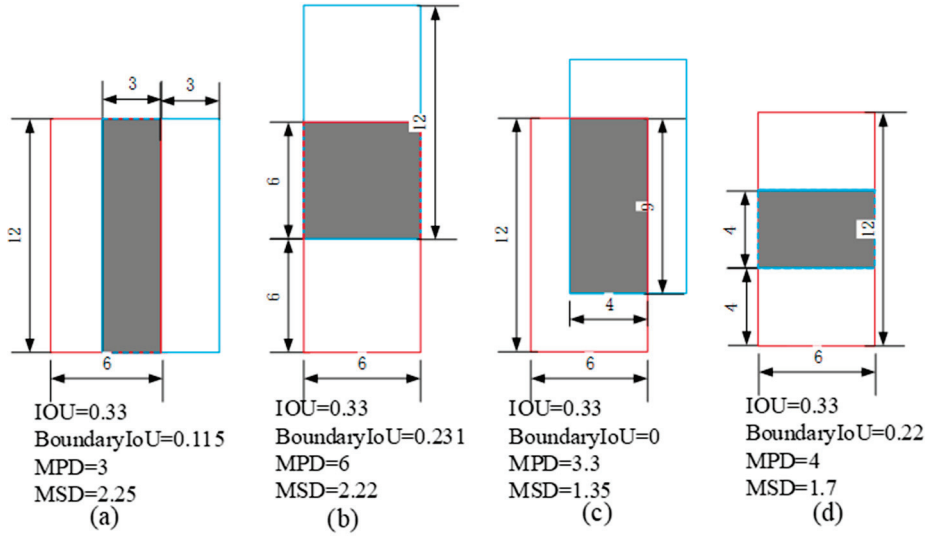


Figure 2. There are several scenarios where IoU values are equal. The red boxes in (**a**–**d**) represent the same reference contour with equal IoU values. The blue boxes indicate the contours to be evaluated, where (**a**–**c**) are the same but have different positions. BoundaryIoU represents the intersection over a union of the extracted contour and the reference contour within a particular buffer zone of distance d. Here, we take d as the minimum value. The overlap between the two contours is represented by red and blue dashed lines. In (**c**), there are only two overlapping points so the BoundaryIoU value is 0.

The mean surface distance (MSD), which is the average of the surface distances of all points in two comparison objects A and B, is a common method for conducting a deep learning accuracy evaluation in medicine. This metric is also known as the average symmetric surface distance (ASSD) [27], which is used in medical image segmentation. It can be used to calculate the average surface distance (ASD) [36] from the predicted segmented object to the reference truth under default settings. It represents the extent to which the surface varies between segmented and reference objects and the collection of pixels at the edge of the surface. The shortest distance to any boundary pixel is defined as

$$d(v, S(A)) = \min_{S_A \in S(A)} \|v - s_A\|, \tag{2}$$

where $\|.\|$ denotes the Euclidean distance. The ASSD can then be obtained by using the following equation:

$$MSD = \frac{1}{|S(A)| + |S(B)|}\left(\sum_{s_A \in S(A)} d(s_A, S(B) + \sum_{s_A \in S(B)} d(s_B, S(A))\right) \tag{3}$$

Distance refers to the shortest distance from a point on the contour to the reference contour—that is, between the distances traversed by the top-point and each of the top-points, pick the smallest one. Although the MSD has simple methods and is widely used for deep learning accuracy evaluation in medicine, it has irreparable defects, as shown in Figure 3.

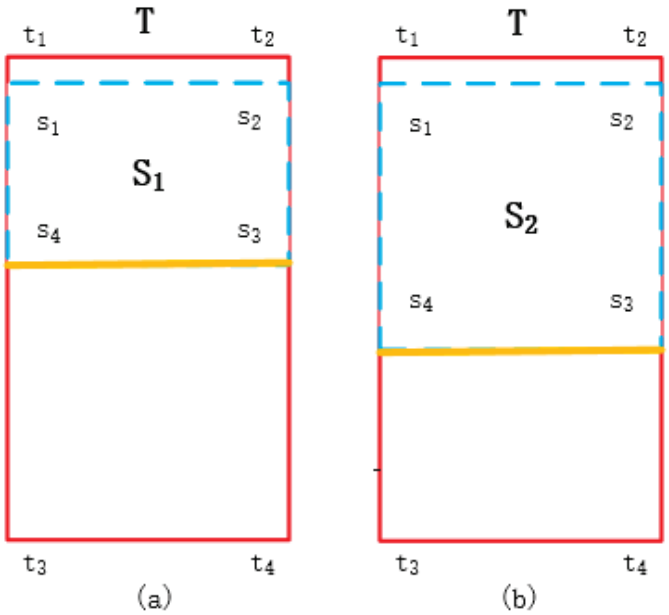


Figure 3. The MSD of (**a**,**b**) are equal, but the contour of (**b**) is significantly better. The shortest distances between the points on S_3S_4 (yellow line) and the reference contour T are the distances between the points on edge S_3S_4 and point (S_3 or S_4), respectively, which are both on edge S_3S_4; thus, $MSD(S_1, T) = MSD(S_2, T)$. An error is added when calculating the average distance between two contours. This is because, when calculating the distance between two contours, the closest distance from the point to the reference contour is used to approximate the distance from the point to the corresponding edge, ignoring the correspondence between the contours and the edges.

The Hausdorff distance [37] measures the extent to which each point of a model set lies near some point of an image set. Given two finite point sets, $A = \{a_1, a_2, \cdots, a_p\}$ and $B = \{b_1, b_2, \cdots, b_q\}$, the Hausdorff distance is defined [38] as:

$$H(A, B) = \max(h(A, B), h(B, A)), \tag{4}$$

$$h(A, B) = \max_{a \in A} \left\{ \min_{b \in B} \|a - b\| \right\}, \tag{5}$$

$$h(B, A) = \max_{b \in B} \left\{ \min_{a \in A} \|b - a\| \right\}, \tag{6}$$

where $\|.\|$ is the underlying norm of the points of A and B (e.g., the L2). $h(A, B)$ is the directed Hausdorff distance from A to B. It identifies the point $a \in A$ that is farthest from any point of B and measures the distance from a to its nearest neighbor in B; that is, $h(A, B)$, in effect, ranks each point of A based on its distance to the nearest point of B and then uses the largest ranked point as the distance. The Hausdorff distance $H(A, B)$ is the maximum of $h(A, B)$ and $h(B, A)$.

As shown in Figure 4, for the two contours X and Y, $H(X, Y) = d_{YX}$ is the maximum distance between the two contours. However, using this indicator to measure the average distance between two contours, assuming that d_{YX} is much larger than the distance between the other points of the contours, causes a large error, and thus it cannot correctly measure the distance between two contours.

Evaluation methods based on IoU cannot calculate the error distance or medium error value of the mapping, and the evaluation results are not necessarily correct because the theoretical design of the calculation method and loopholes is insufficient. Therefore, there is an urgent need to study and design a theoretical metric system suitable for evaluating the accuracy of artificial intelligence vector images.

The contributions of this study are as follows:

(1) An inflection point matching algorithm of a vector polygon is proposed to match the inflection points of the extracted building contour and the reference contour.
(2) We define and formalize the edge inflection point of the vector contour. The inflection points extracted by artificial intelligence and the inflection points of ground truth do not always correspond one-to-one. If these inflection points are not distinguished, it will significantly increase the evaluation error. Experiments have shown that it is important to clearly distinguish the inflection points on the contour.
(3) Mean inflection point distance (MPD) is calculated and realized, and its effectiveness is verified via experiments.

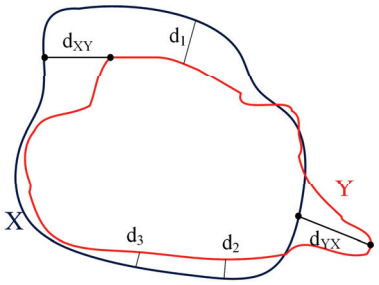


Figure 4. Two contours, where red is Y and blue is X. d_{XY} is the maximum distance from X to Y. d_{YX} is the maximum distance from Y to X, because $d_{XY} < d_{YX}$. Therefore, $H(X, Y) = d_{YX}$.

2. Methods

Figure 5 depicts the MPD procedure. First, data preprocessing is performed, which includes one-to-one correspondence between the extraction results and the individual instance mask of the ground truth. The extraction mask is binary, regularized, and vectorized, and the contour inflection points are extracted. After that, the extracted and reference inflection points are matched. Second, to evaluate the accuracy of our approach, we conducted two different scenarios: non-discrimination of edge inflection points (MPD_EP), which calculates the average distance of the corresponding inflection points, and discrimination of edge inflection points (MPD). In the latter scenario, if there are edge inflection points, corresponding inflection points are added to the corresponding edge to generate new pairs of corresponding inflection points. We then calculate the average distance of these corresponding inflection points.

2.1. Data Preprocessing

Artificial intelligence-based accuracy evaluation begins with data preprocessing; binarization, regularization, and vectorization of the extracted spots; extraction of inflection points and boundary line segments; and inflection matching of the extracted contours and reference contours. The data preprocessing process is shown in Figure 5.

1. Mask one-to-one correspondence: The extracted building spots and reference spots on each image are matched one-to-one. For example, we used IoU and calculated IoU for one extracted spot and all reference spots on each image. The two spots with the largest value that is greater than a certain threshold (for example, we chose 0.5) correspond to each other.
2. Binarization: Binarize the extracted RGB masks to obtain grayscale masks.
3. Contour extraction: Extract contours from the binarized mask. For example, we used the findContours [39] algorithm to extract the contours.
4. Regularization: The vector contour is regularized. For example, we used the Douglas–Peucker (DP) [40] algorithm to obtain the inflection points of the extracted contours. Inflection points of the reference contour were obtained using manual annotations.
5. Inflection point matching: We used the dynamic programming [41] inflection point matching algorithm to obtain matched pairs of inflection points. The inflection point

matching problem was simplified to extract m-ordered inflection points $p_i(i = 1, 2, \cdots, m)$ of the contour A. Next, A was divided into m segments. The reference contour B had q inflection points $q_j(j = 1, 2, \cdots, n)$. Thereafter, B was divided into n segments. The inflection point matching of two contours was the multivalued mapping from inflection point set a_i of A to inflection point set b_i of B of $i : j \rightarrow \{i(j)\}$, where $i = 1, 2, \cdots, m, i(j) = 1, 2, \cdots n$. There were three cases of inflection point matching of two polygon contours: one-to-one, one-to-many, and many-to-one. All inflection points are required to participate in matching; therefore, they must be surjective, meaning that the inflection point matching of the two contours is transformed into a target optimization problem.

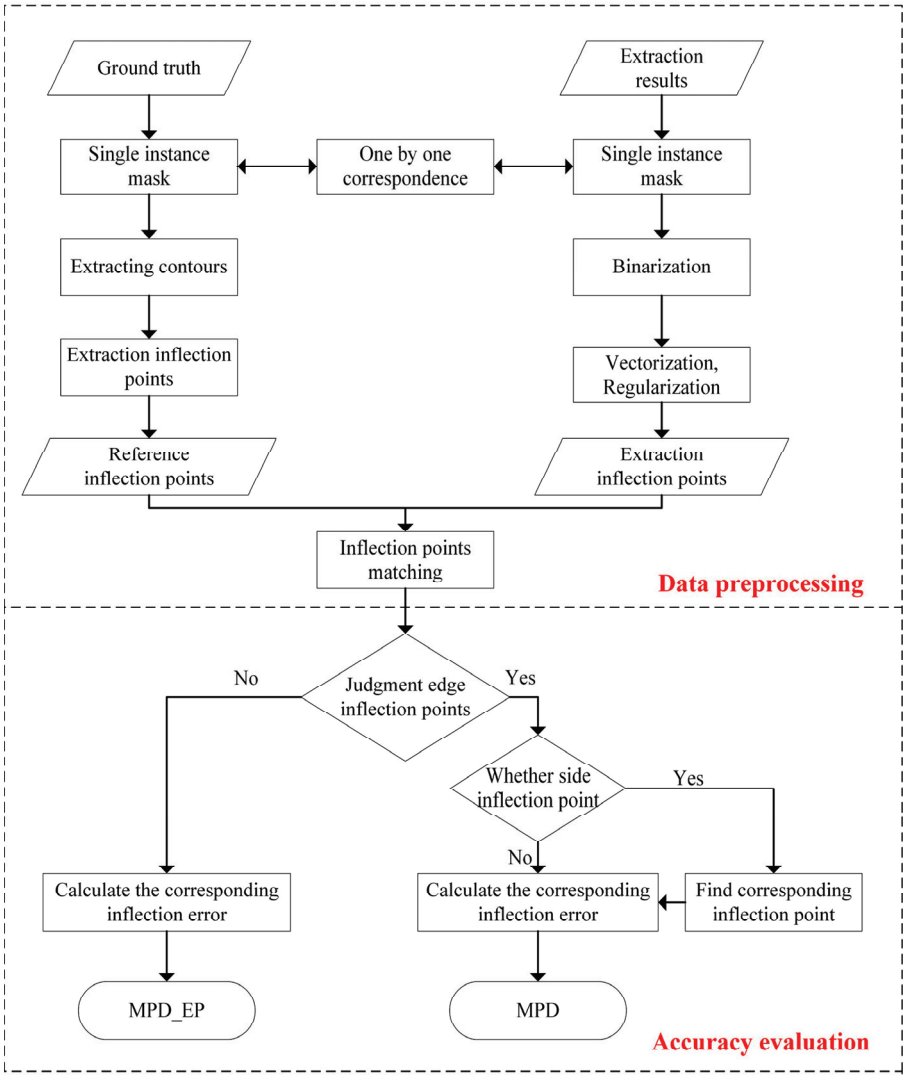


Figure 5. Overall flowchart of MPD.

As shown in Figure 6b, contour S and reference contour T were extracted for inflection point matching, and the best matching route is shown via the red curve in Figure 6a. The optimal match of the two contour inflection points can be determined by the line from the bottom row in the dynamic programming table. The path to the top row passes through the smallest sum of the grid values. The path can be extended horizontally or vertically, indicating a one-to-many or many-to-one path. Multiple values in the same row indicate that the extracted contour S had one inflection point corresponding to many inflection points of the reference contour T; multiple values in the same column indicate that the extracted contour S had multiple inflection points corresponding to one inflection point

of the reference contour T. In Figure 6b, the inflection points that match contour S and reference contour T were extracted, and the best matching route is indicated by the red curve in Figure 6a. S and T had five inflection points each, matching the six inflection point pairs.

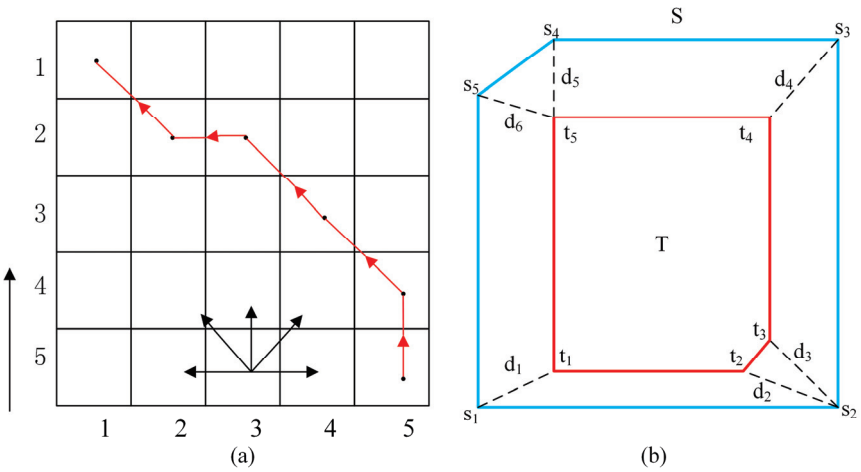


Figure 6. Inflection point matching. (**a**) This is a DP table, and a path in the table can be represented as follows: starting from the bottom row with a certain direction (left or right), moving to the grid with the minimum value in the DP table one grid at a time, and ending at the vertex row. The red line in (**a**) represents the matching result of the inflection points of two contours in (**b**). (**b**) The inflection points matching contour S and reference contour T are extracted, and the best matching route is shown in the red curve in (**a**). S and T each has five inflection points, matching six inflection point pairs.

2.2. Mean Inflection Point Distance

The number of inflection points of the target contour extracted by the intelligent method and the reference contour may not be the same in the accuracy evaluation; the inflection points of the extracted contour and reference contour are S and T, respectively. The inflection point matching algorithm was used to match the two contour inflection points, and $s_i \rightarrow t_j$ generated k inflection point pairs. As shown in Figure 5b, the extracted contour S and reference contour T each have five inflection points, and matching produces six inflection point pairs. The distance $d_k = \|s_i, t_j\|$ matches the inflection point. The MPD_EP (mean inflection point distance of the indistinguishable edge inflection points) is calculated as follows:

$$\text{MPD_EP} = \frac{\sum d_k}{k} \tag{7}$$

The matching inflection points of a few building contours are distant, but proximate to the matching edge. These inflection points are called edge inflection points, and the rest are called vertex inflection points (as shown in Figure 7). The inflection point of the extracted contour is far from the corresponding inflection point of the reference contour, and the distance to the corresponding edge is minuscule. Therefore, the inflection point is an edge inflection point. If it is an edge inflection point, calculating the inflection point distance with the matching inflection point cannot truly reflect the mapping accuracy; therefore, the matching inflection point of the edge inflection point is the base of its corresponding edge. There are three conditions to determine the inflection point:

(1) The inflection point must be 1-to-n or n-to-1. If it is 1-to-1, there is no edge inflection point;
(2) The base from the inflection point to the corresponding side is on the corresponding side;
(3) The distance from the inflection point to the matching inflection point satisfies the following formula:

$$d(s_i, t_j) > \alpha * MPD, \tag{8}$$

where $d(s_i, t_j)$ is the distance to the corresponding inflection point and α is the judgment coefficient, such as 2–5.

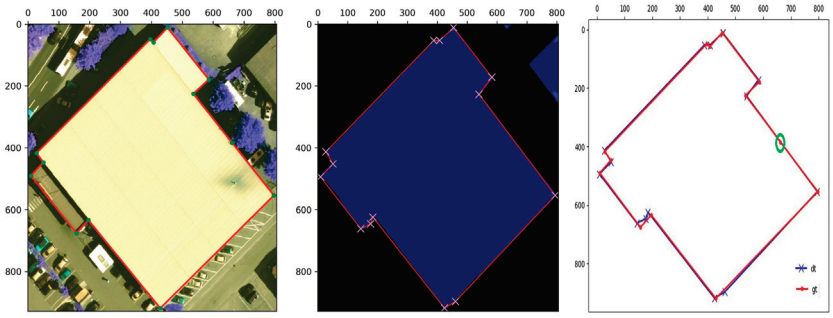


Figure 7. Contours with edge inflection points. (**a**) Original image and reference contour; (**b**) extracted mask and regularized contours; (**c**) two overlapping contours. The inflection point in the green circle is very close to the edge, and the corresponding inflection point is a large distance away; thus, the point is an edge inflection point.

If the distance between the inflection point and the corresponding edge meets the conditions, select any of the following formulas according to different actual conditions:

$$\begin{cases} \min(d(s_i, t_j t_{j+1} | t_{j-1} t_j)) < m | \\ \min(d(s_i, t_j t_{j+1} | t_{j-1} t_j)) < \beta * MPD \end{cases}, \tag{9}$$

where $d(s_i, t_j t_{j+1} | t_{j-1} t_j)$ is the distance between the contour inflection point and the two edges containing the corresponding inflection point; m is the number of specified pixels, such as, for example, 5–10; β is the judgment coefficient, such as, for example, 0.5–2; and λ is the judgment coefficient, which can be 0.5–2.

The MPD can be calculated by the following formula:

$$MPD = \begin{cases} \frac{\sum_{i=1}^{k-b} d(s_i, t_j) + \sum_{i=1}^{b} \min(d(s_i, t_j t_{j+1} | t_{j-1} t_j))}{n+b}, m \geq n \\ \frac{\sum_{i=1}^{k-b} d(s_i, t_j) + \sum_{i=1}^{b} \min(d(s_i, t_j t_{j+1} | t_{j-1} t_j))}{m+b}, m < n \end{cases}, \tag{10}$$

where k is the number of matching pairs of inflection points, b is the number of edge inflection points, and m and n are the numbers of inflection points of the extracted contour and reference contour, respectively.

3. Experiment and Analysis

3.1. Experimental Setting

3.1.1. Experimental Hardware and Software Environment

The network model was built using PyTorch 1.6, Python 3.7, and CUDA 10.0. The training and testing of the experiment were conducted on an Ubuntu 16.04 server with an Intel(R) Xeon(R) CPU E5-2650 v3, 32 GB RAM, and an Nvidia RTX 2080 GPU*2. We used the Adam optimizer with 0.9 momenta. The initial learning rate was 0.001, the batch size in the training phase was fixed at 2, and the image cutoff was 1792 × 1792. The network was trained for 50 epochs. Three deep learning methods were adopted in this study: PointRend [42], Swin Transformer [43], and Mask R-CNN [44]. All models used the MMDetection deep learning framework.

3.1.2. Description of Experimental Data

We used the International Society for Photogrammetry and Remote Sensing (ISPRS) benchmark of Vaihingen [45], which is an open benchmark dataset for extracting various urban targets, including buildings. The dataset contained three false-color aerial images (near-infrared, red, and green channels) with a ground resolution of 9 cm.

3.1.3. Experiment Design

We used buildings as objects of interest because of their clear boundaries and representative shapes and sizes to more objectively compare mapping accuracy results and better track the performance of the metrics [46]. To measure the reference contours corresponding to the extracted contours more rationally, we calculated metrics based on the reference contour polygons and their corresponding extracted contours, and then analyzed their performance relative to the best segmentation determined by visual inspection to examine the sensitivity of each metric to the mapping accuracy. Two different phases were designed for the experiments. In the first stage, three groups of buildings with different complexities in shape/construction (simple, fairly complex, and complex) were selected, with each group varying in size, shape, color, and roof complexity. In the second step, we evaluated the overall accuracy of 800 buildings, and the reference contours of all buildings were obtained by manual annotation.

3.2. Evaluation Metrics

Five evaluation metrics were employed for quality evaluation in this research: overall accuracy (IoU) (Equation (1)), Boundary IoU (Equation (11)), MSD (Equation (3)), MPD_EP (Equation (7)), and MPD (Equation (10)). Boundary IoU is calculated as follows:

$$Boundary\ IoU(G, P) = \frac{|(G_d \cap G) \cap (P_d \cap P)|}{|(G_d \cap G) \cup (P_d \cap P)|}, \tag{11}$$

where G is the ground truth mask; P is the prediction mask; and G_d and P_d are sets of pixels in the boundary region of the binary mask.

3.3. Accuracy Evaluation of Buildings with Different Complexities in Shape/Construction

As shown in Figure 8, the buildings were rectangular/square, and the results extracted by the three deep learning methods were consistent with the reference contour with regard to a few edge inflection points. The experimental results are shown in Figure 8 and Table 1. The IoUs of three buildings extracted by the three algorithms (PointRend, Swin Transformer, and Mask R-CNN) exceeded 92%. The Boundary IoU values were all greater than 85%, while the MSD values were only larger than 5 when using the Mask R-CNN method for building a, and the quality of mask extraction was excellent. For building a, PointRend was the best among all the accuracy metrics. The MPD values for the three algorithms were 3.15, 6.40, and 8.34, and the MPD_EP values were 20.66, 27.71, and 37.68, respectively. Because of the existence of edge inflection points, the MPD_EP was more than four times the MPD, which fully demonstrates that it is essential to determine the edge inflection points. For building b, the mask metrics of PointRend had high accuracy; however, the contour accuracy of Swin Transformer was high. The MSD values of the three algorithms were 1.89, 2.58, and 2.39, and all were very small compared to the MPD values. As previously analyzed, the MSD does not consider the corresponding relations between edges when calculating distance; thus, the calculation will contain errors and cause the results to be smaller. The MPD_EP is significantly larger than the MPD when there are edge inflection points; otherwise, the MPD_EP equals the MPD.

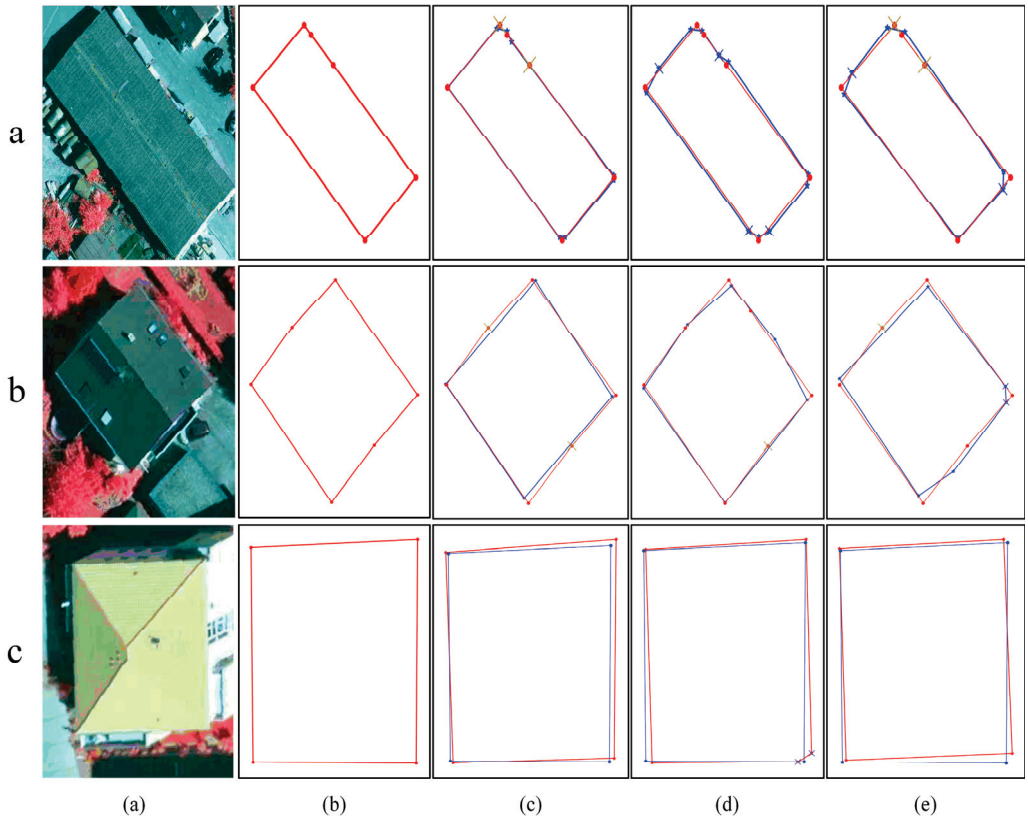


Figure 8. Contour extractions for buildings with simple structures. (**a**–**e**) are respectively the original image, ground truth, PointRend, Swin Transformer, and Mask R-CNN. The red line represents the ground truth, the blue line represents the extracted contour, and X denotes the edge inflection points of the vector contour.

Table 1. Evaluation metrics of contour extraction effects for buildings with simple structures.

Building	Method	IoU	Boundary IoU	MSD	MPD_EP	MPD
a	PointRend	97.38%	94.11%	2.83	20.66	3.15
	Swin Transformer	94.82%	87.65%	4.58	27.71	6.40
	Mask R-CNN	94.35%	87.16%	5.01	37.68	8.34
b	PointRend	95.59%	95.07%	1.89	28.75	4.41
	Swin Transformer	93.94%	92.81%	2.58	24.51	3.97
	Mask R-CNN	94.51%	93.73%	2.39	21.99	9.38
c	PointRend	94.61%	90.84%	2.68	5.39	5.39
	Swin Transformer	92.73%	86.69%	3.68	6.34	4.59
	Mask R-CNN	93.62%	88.06%	3.23	5.89	5.89

As shown in Figure 9, the building contours were much more complex in structure, mostly irregular, and had many folded corners. The contours extracted by the three algorithms can be seen in many places to be not accurate enough; for example, the corners of building a were not extracted well, and many corners on building b were not extracted by Swin Transformer and Mask R-CNN, resulting in a line. The extraction results of building c were better in PointRend overall, but the lower right part exceeded the ground truth.

The experimental results are shown in Figure 9 and Table 2. For building a, the IoUs of the extracted masks by the three algorithms (PointRend, Swin Transformer, Mask R-CNN) were all lower than 90%; moreover, the Boundary IoUs were less than 75%. The accuracy of the masks extracted by the three algorithms was not high. For the accuracy evaluation of the extracted contours, the MPD_EP values were 26.45, 39.67, and 47.69; all MPD values

were greater than 17; and the visual effect was consistent. Thus, all the evaluation index results show the low accuracy of the extracted contours. For building b, the IoUs of the three algorithms' extracted masks were greater than 90%; the Boundary IoUs were greater than 87%; PointRend's extraction results of corner and ground truth matched the best; and the MPD values were the lowest. For building c, the IoUs of the three algorithms' extracted masks were 95.36%, 93.81%, and 91.05%; the Boundary IoUs were 90.59%, 82.62, and 76.47%; and the MPD values were 20.31, 18.91, and 32.97. Because there was a protruding part in the lower right corner of the contours extracted by PointRend, the MPD values of PointRend were larger than those of Swin Transformer.

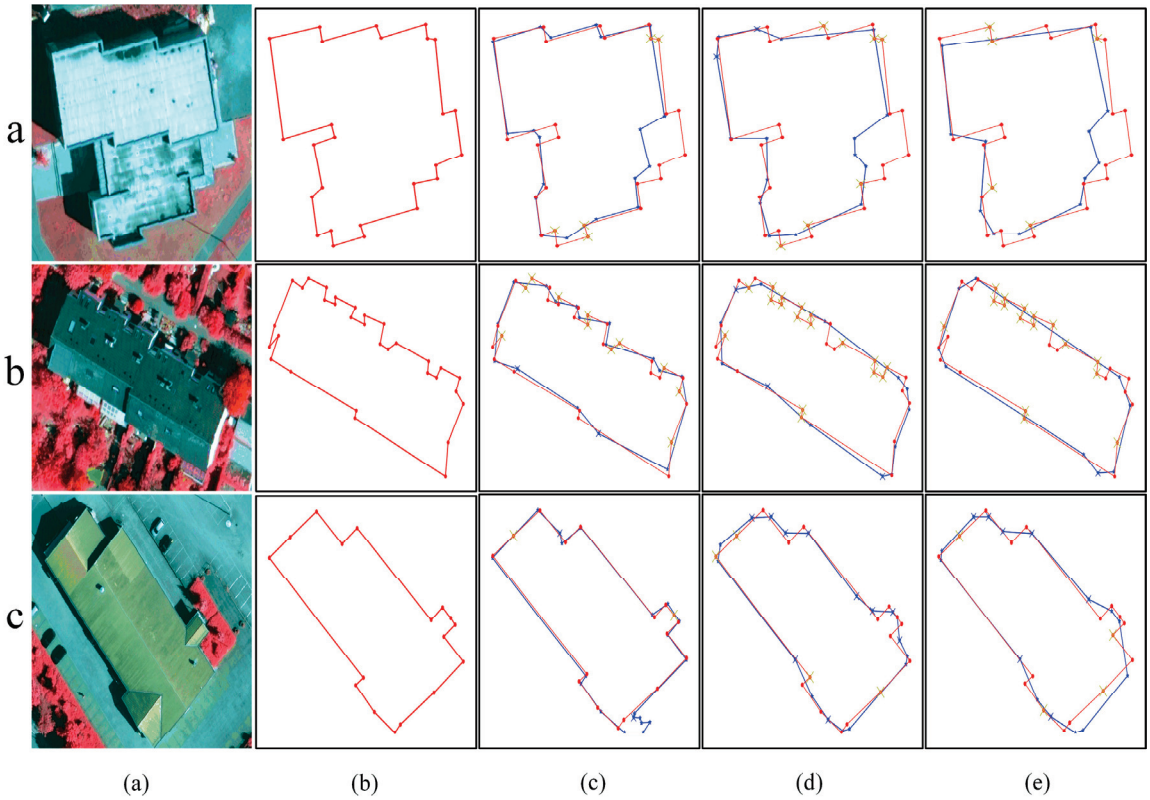


Figure 9. Contour extractions for buildings with more complex and irregular structures. (**a**–**e**) are respectively the original image, ground truth, PointRend, Swin Transformer, and Mask R-CNN. The red line represents the ground truth, the blue line represents the extracted contour, and X denotes the edge inflection points of the vector contour.

Table 2. Evaluation metrics of contour extraction effects for buildings with more complex and irregular structures.

Building	Method	IoU	Boundary IoU	MSD	MPD_EP	MPD
a	PointRend	85.56%	73.15%	6.97	26.45	17.12
	Swin Transformer	86.33%	69.30%	8.47	39.67	19.44
	Mask R-CNN	85.61%	69.08%	8.83	47.69	25.23
b	PointRend	94.94%	91.40%	2.80	15.74	8.84
	Swin Transformer	92.30%	87.11%	4.10	31.48	11.70
	Mask R-CNN	92.38%	87.12%	3.99	64.80	12.93
c	PointRend	95.36%	90.59%	4.78	31.71	20.31
	Swin Transformer	93.81%	82.62	5.75	36.02	18.91
	Mask R-CNN	91.05%	76.47%	8.50	39.16	32.97

The structures of the buildings presented in Figure 10 were much more complex, dense, and mostly irregular; thus, significant and clear errors were observed in many places in the contours extracted by the three algorithms (Figure 10). The corners of building a were not extracted well, and the building details were not extracted. For building b, Swin Transformer and PointRend had multiple mentions of the parts. For building c, all three algorithms extracted the adjacent building next to the main point of interest, and the reference contour had very large errors. The experimental results are shown in Figure 10 and Table 3. For building a, the mask accuracy extracted by Mask R-CNN was high, as was the contour accuracy after regularization, which is consistent with Figure 10. For building b, the accuracy metric of the three algorithms was not high, whereas for building c, the IoUs of the three algorithms' extracted masks were lower than 70%; the Boundary IoUs were lower than 65%; the MPD values were greater than 28; and the difference between the reference contour and the extracted contour was large. However, the Mask R-CNN results were a bit better. For building c, the performances of the three algorithms were poor, no edge inflection points existed, the MPD and MPD_EP were equal, and the error of the MPD was enormous, exceeding 109. The extraction quality of the mask was poor, and the quality of the regularized contours was equally poor.

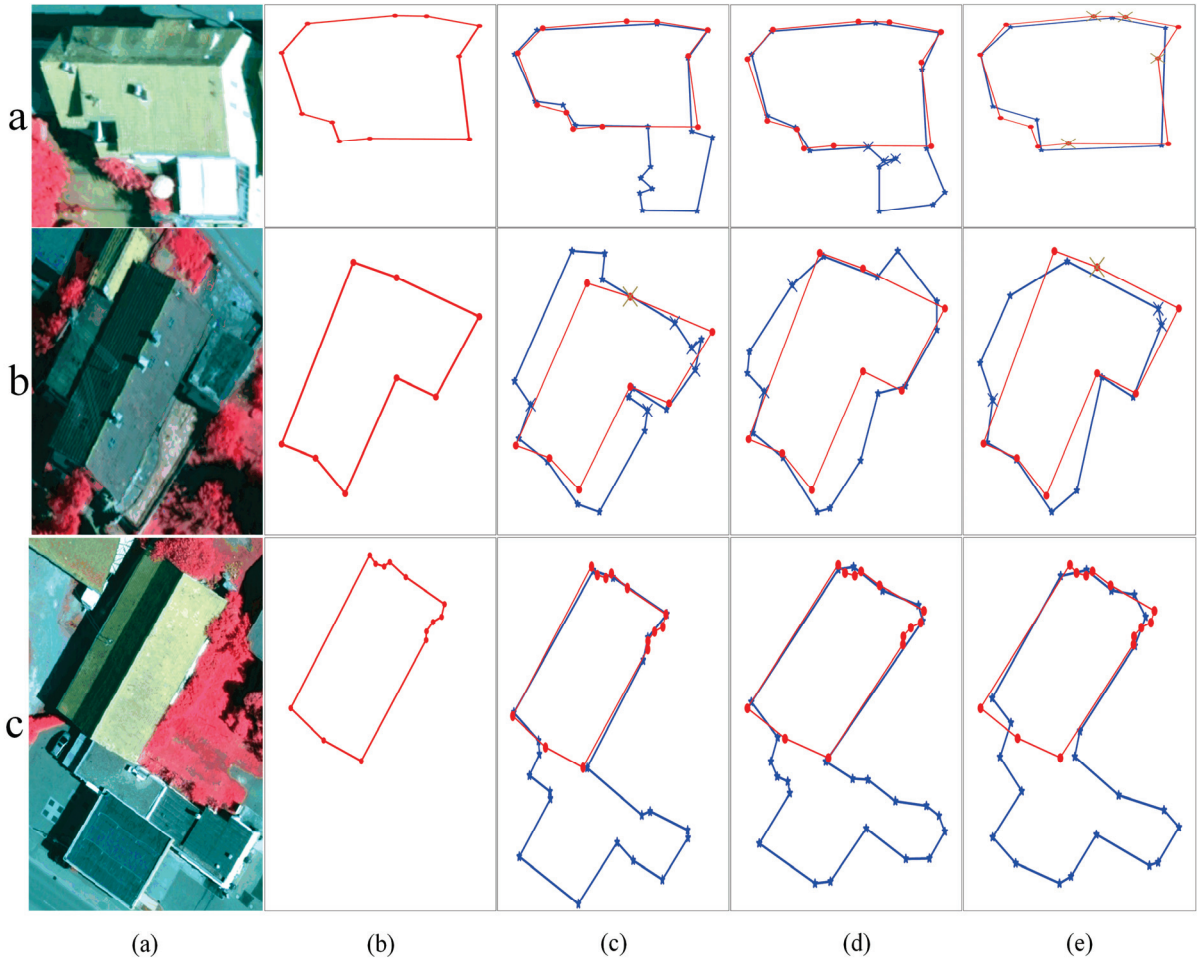


Figure 10. Contour extractions for buildings with complex structures. (**a**–**e**) are respectively the original image, ground truth, PointRend, Swin Transformer, and Mask R-CNN. The red line represents the ground truth, the blue line represents the extracted contour, and X denotes the edge inflection points of the vector contour.

Table 3. Evaluation metrics of contour extraction effects for buildings with much more complex and denser structures.

Building	Method	IoU	Boundary IoU	MSD	MPD_EP	MPD
a	PointRend	72.07%	68.82%	16.48	95.58	47.75
	Swin Transformer	80.81%	77.80%	12.56	32.41	27.04
	Mask R-CNN	90.26%	89.92%	8.53	15.22	8.34
b	PointRend	63.86%	57.61%	18.26	34.79	30.11
	Swin Transformer	69.87%	59.40%	15.76	37.10	34.61
	Mask R-CNN	71.36%	65.33%	14.21	32.56	28.24
c	PointRend	52.52%	47.78%	53.16	109.91	109.91
	Swin Transformer	53.58%	48.60%	51.59	109.23	109.23
	Mask R-CNN	50.26%	44.48%	54.90	109.96	109.96

3.4. Overall Accuracy Evaluation

In the previous subsection, we evaluated the extraction accuracy of nine representative reference buildings to analyze and compare the effectiveness and sensitivity of various metrics. However, in a real project, multiple reference polygons should be evaluated. Therefore, we selected 800 reference buildings with different shapes, colors, sizes, and structures. As listed in Table 4, the three algorithms averaged the accuracy evaluation of the extracted contours; the best results were found for PointRend, the second best were found for Swin Transformer, and the worst were found for Mask R-CNN in each index. The evaluation results of the masks were consistent with the vector evaluation results proposed in this study, which demonstrates the effectiveness of the proposed index for application to buildings of different shapes and complexities. Taking PointRend as the benchmark, the differences in the IoU values of the three algorithms were less than 1%, and the differences in the Boundary IoU values were less than 2%. The two indices do not significantly differentiate the extraction results. The differences between the MPD values of PointRend and Swin Transformer, as well as PointRend and Mask R-CNN, were 1.04 and 2.06, respectively, which translates to 11.44% and 28.60%, respectively; therefore, the MPD has a higher numerical discrimination and is more sensitive to mapping accuracy. The MPD values were significantly smaller than the MPD_EP values, and greater than 10. This fully indicates that the edge inflection point plays a significant role in determining the contour accuracy, and must be distinguished.

Table 4. Averaged evaluation metrics of 800 buildings' extracted contours.

Methods	$IoU_{average}$	Boundary $IoU_{average}$	$MSD_{average}$	$MPD_EP_{average}$	$MPD_{average}$
PointRend	90.75%	89.08%	3.34	19.57	9.09
Swin Transformer	89.75%	87.79%	3.58	21.76	10.13
Mask R-CNN	89.90%	87.49%	3.93	23.49	12.05

3.5. Analysis of Experimental Results

Tables 1 and 2 show that the PointRend algorithm outperformed the other two for extracting simple and slightly complex building outlines. Furthermore, we discovered a strong correlation between the trends revealed by IoU, BoundaryIoU, MSD, and MPD. BoundaryIoU increased as IoU increased, MSD decreased, and MPD decreased.

In Table 3, we discovered that the PointRend algorithm had a small MSD but a large MPD for building c. We discovered this through image analysis because the extracted corner points were concentrated in the lower right corner and had a large error, and the MPD was 109.91. As a result, MPD produced biased evaluation results, which is a disadvantage of this metric. The evaluation results will be even more biased if the discrete corner points are too concentrated and have a large error. Because MPD EP is much greater

than MPD when there are edge corner points, distinguishing and identifying edge corner points is critical.

Finally, we extracted a large number of buildings in Table 4. Under extensive sample conditions, we discovered that IoU, BoundaryIoU, MSD, and MPD all had a strong correlation, demonstrating the effectiveness of MPD as an accuracy metric.

To summarize, all indicators performed well when building structures were simple and uncomplicated. However, when building structures were complex, all indicators had limitations, with low extraction accuracy. The advantage of MPD is that it can calculate the actual value error; however, it can have biases when discrete corner points are over-configured.

4. Discussion

4.1. Comparison with the Networks That Predict Buildings

We compared the results of PointRend [42], Swin Transformer [43], and Mask R-CNN [44] both quantitatively and qualitatively. Mask R-CNN and Swin Transformer often predicted labels on a low-resolution regular grid, e.g., 28×28 [47], for instance, segmentation, as a compromise between undersampling and oversampling. PointRend already achieves excellent results. Qualitatively, PointRend outputs crisp object boundaries in regions that are oversmoothed by Mask R-CNN. Quantitatively, PointRend yields significant gains on COCO and Cityscapes for both instance and semantic segmentation. For making inferences with predicted buildings, we used the adaptive subdivision technique to refine the coarse 7×7 prediction for class c to reach 448×448 in six steps. At each step, we selected and updated (at most) the $N = 282$ of the most uncertain points. PointRend improved the resolution by iteration to obtain a high-resolution mask (as shown in Figure 11).

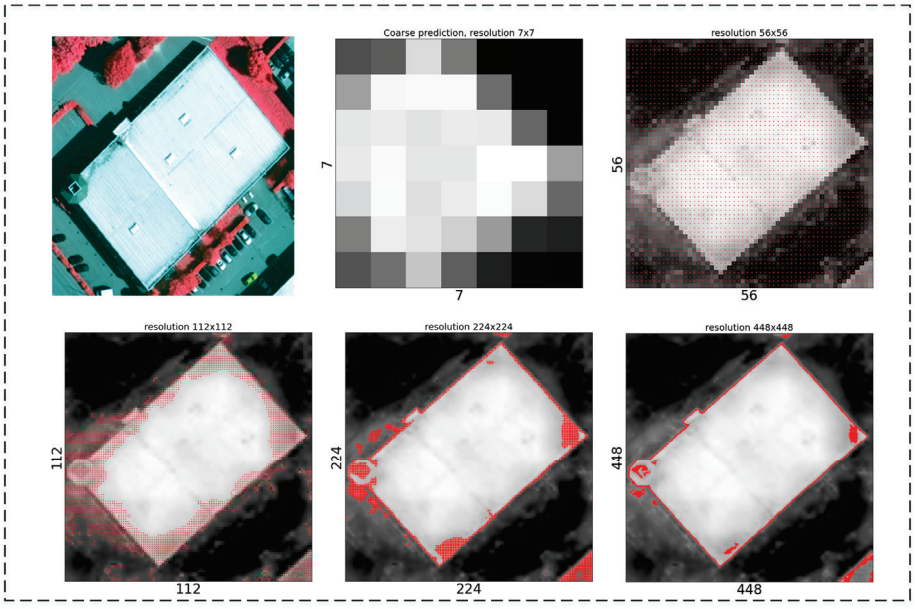


Figure 11. Example of one adaptive subdivision step. The red dots are the points that need to be inferred for each iteration.

Figure 12 shows the comparison results for the dataset. From top to bottom, the buildings in the first image are more regular and smaller, and all three algorithms had better extraction results. The second image has buildings with complex structures and many folded corners (shown in yellow boxes), and PointRend extracted the highest building integrity, showing sharper corners. The third image has a large building area, and the edges of the building patches extracted by Swin Transformer and Mask R-CNN show jagged edges and blurred borders. The buildings in the fourth image are very dense, and all three algorithms extracted multiple buildings as one building. To better clarify the detailed inspection, the close-ups of the selected buildings (as marked in yellow rectangles

in Figure 12) from the tested images are displayed in Figure 13. From the close-up views, we can observe that PointRend was significantly better at extracting buildings with complex structures and many corners than Swin Transformer and Mask R-CNN. For buildings with a relatively large area, Swin Transformer and Mask R-CNN had obvious jagged shapes on the edges of the extraction results.

Figure 14 shows an overlay comparison of the building extraction and ground truth results obtained with different methods. The close-ups of the selected regions (as marked in red rectangles in Figure 14) in the tested images are displayed in Figure 15. For buildings with a simple structure, regular shape, and small size, the extraction results of the three algorithms are not that different; for buildings with a complex structure, such as in the second and third pictures in Figure 15, the extraction results of PointRend are significantly better than those of Swin Transformer and Mask R-CNN. The buildings in the fourth image are too dense, and all three algorithms extracted multiple buildings as one, showing a similar performance.

Based on the above analysis, we can conclude that the improvement in the performance of PointRend lies in the identification of multiscale buildings, especially large-scale and structurally complex buildings, as well as the more accurate positioning of building boundaries.

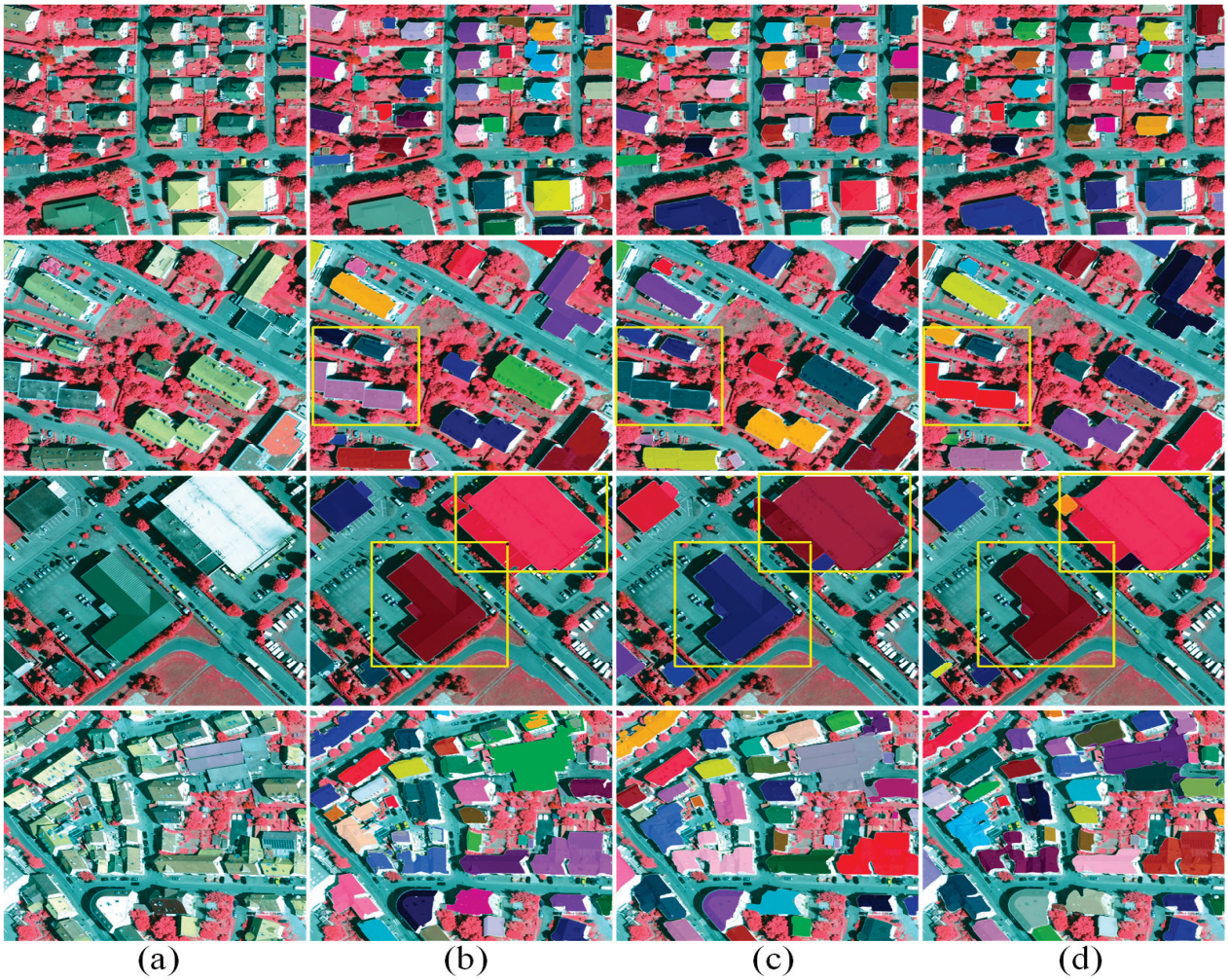


Figure 12. Examples of building extraction results obtained by different methods. (**a**) Original image. (**b**) PointRend. (**c**) Swin Transformer. (**d**) Mask R-CNN. The yellow rectangles in (**b**–**d**) are the selected buildings for close-up inspection, shown in Figure 13.

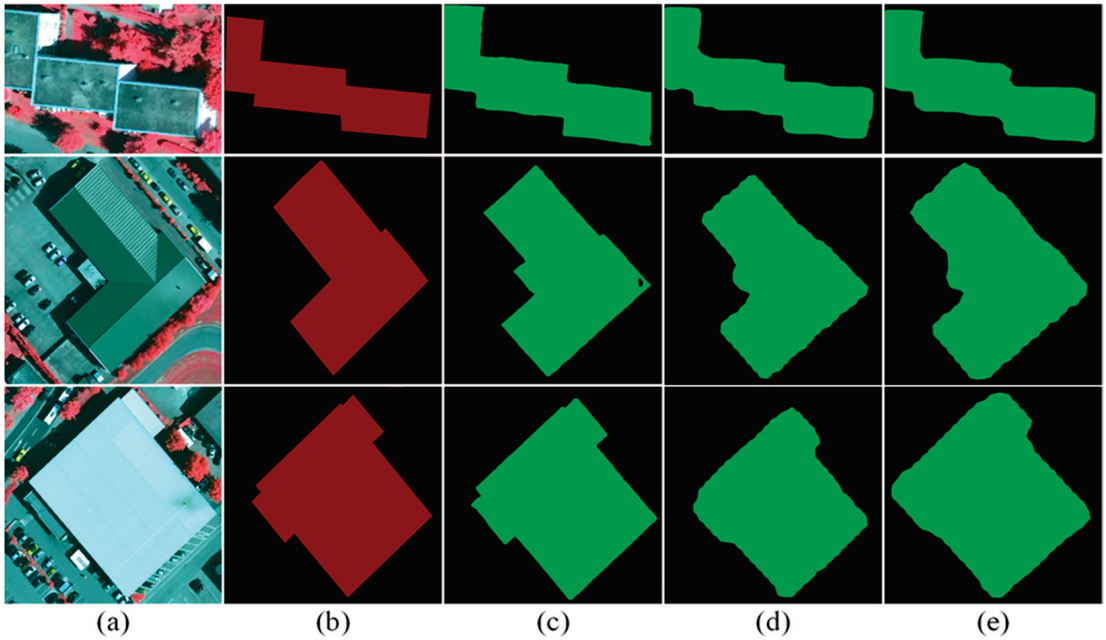


Figure 13. Close-up view of individual building results obtained by different methods. (**a**) Original image. (**b**) Ground truth. (**c**) PointRend. (**d**) Swin Transformer. (**e**) Mask R-CNN.

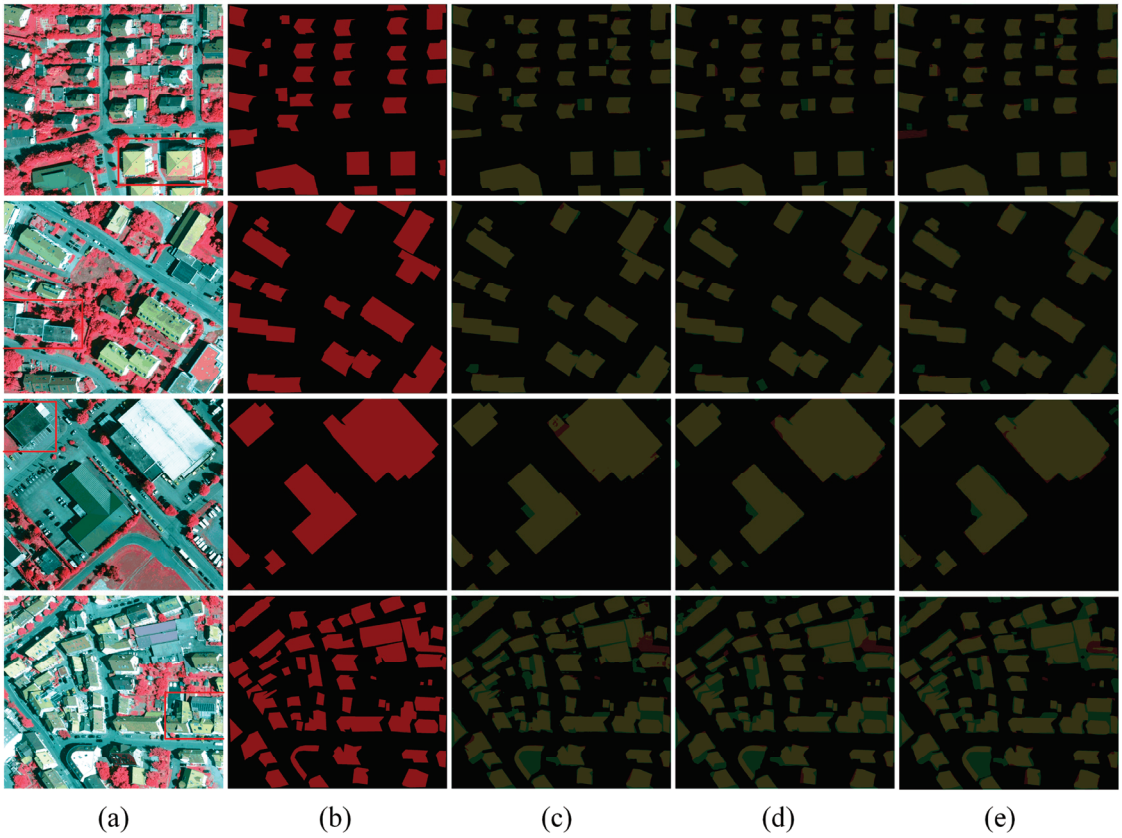


Figure 14. Examples of building extraction results obtained by different methods. (**a**) Original image. (**b**) Ground truth. (**c**) PointRend. (**d**) Swin Transformer. (**e**) Mask R-CNN. Note, in columns (**c**–**e**), yellow, green, and red indicate true-positive, false-negative, and false-positive, respectively. The red rectangles in (**a**) are the selected regions for close-up inspection, shown in Figure 15.

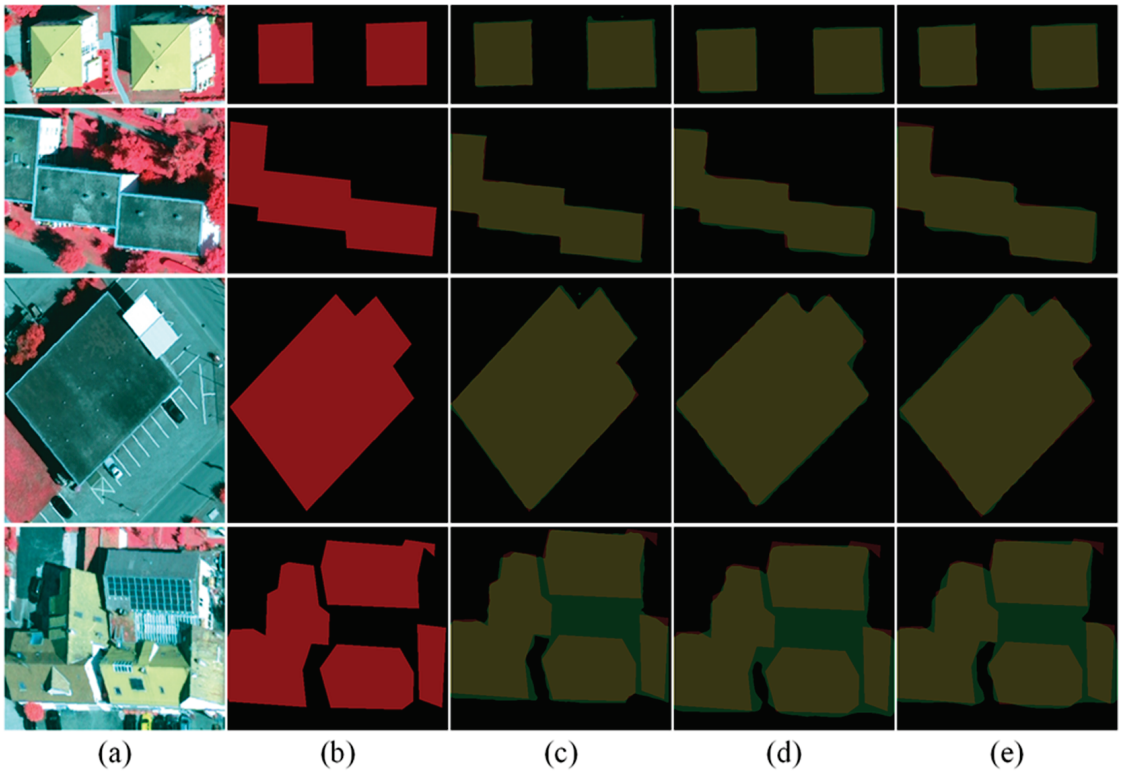


Figure 15. Close-up views of the results obtained by different methods. Images and results shown in (**a**–**e**) are the subset from the selected regions marked in Figure 9a. (**a**) Original image. (**b**) Ground truth. (**c**) PointRend. (**d**) Swin Transformer. (**e**) Mask R-CNN.

4.2. More Experiments with Different IoU and MPD Thresholds

In the instance segmentation task, the standard MS COCO metric [22] was used to evaluate the task, which included the mean average precision (AP) of the joint of multiple IoU values. The IoU was given by the following equation:

$$IoU = \frac{area(dt \cap gt)}{area(dt \cup gt)}, \tag{12}$$

where dt and gt denote the predicted mask and the corresponding ground truth, respectively. TP and FP can be determined based on different thresholds. For example, IoU and MPD were used as thresholds. In this paper, we chose MPD thresholds of 3, 7, and 10.

$$P = \frac{TP}{TP + FP}, \tag{13}$$

where TP is the count of true predictions for the buildings and FP is the count of false predictions for the buildings.

Specifically, AP was calculated in steps of 0.05 at 10 IoU overlap thresholds (0.50 to 0.95) [20]. AP@0.5: 0.95 was calculated according to Equation (14).

$$AP_{@0.5:0.95} = \frac{AP_{0.5} + AP_{0.55} + \ldots + AP_{0.9} + AP_{0.95}}{10}, \tag{14}$$

The IoU was used to measure how much our predicted contour overlapped with the ground truth. In this section, because different IoU thresholds produce different levels of accuracy, multiple IoU values and MPD values were used to observe the performance of different approaches. The best results are shown in bold in Table 5. Taking IoU as the threshold, Swin Transformer had the best performance, and the AP@0.5:0.95 of the Swin Transformer exceeded that of PointRend and Mask R-CNN, which were 0.041 and 0.027,

respectively. At IoUs of 0.5 and 0.75, Swin Transformer's AP always achieved the best results, while PointRend's results were the worst. PointRend performed best when the MPD was used as a threshold. When the MPD (unit: pixel) was 3, AP@3 exceeded 0.051 and 0.217 for Swin Transformer and Mask R-CNN, respectively, and was much larger than that of Mask R-CNN. PointRend's AP obtained the best results, with MPDs of 7 and 10. When IoU was the threshold, Swin Transformer had the best performance, whereas when MPD was the threshold, PointRend performed the best, which is consistent with the MSD values and visual results. PointRend enhanced the image edge through iteration, and thus its boundary accuracy was the highest. Therefore, what kind of evaluation should we take? Taking mapping as an example, firstly, the mask extracted by deep learning methods was vectorized and regularized. More attention was paid to the accuracy of the vector boundary, and the mean square error was controlled according to the different scale requirements. The experimental results showed that the metrics proposed in this paper are sensitive to the mapping accuracy, consistent with the MSD value, and can calculate an accurate error value, which is highly consistent with the visual effect. The related research in this paper provides a good reference for carrying out the evaluation of the accuracy of remote sensing artificial intelligence automatic mapping.

Table 5. Comparison results with different thresholds and methods.

Methods	$AP_{IoU@0.5:0.9}$	$AP_{IoU@0.5}$	$AP_{IoU@0.75}$	MSD	$AP_{MPD@3}$	$AP_{MPD@7}$	$AP_{MPD@10}$
PointRend	0.584	0.808	0.660	3.34	0.328	0.500	0.747
Swin Transformer	0.625	0.874	0.709	3.58	0.277	0.405	0.642
Mask R-CNN	0.598	0.848	0.672	3.93	0.011	0.300	0.549

4.3. Accuracy Analysis of HD Metrics

Table 6 shows the results of an evaluation of the HD accuracy of building contours extracted by three algorithms: Mask R-CNN, Swin Transformer, and PointRend. The error values of all three algorithms exceeded the MPD index. Figure 16 shows that the error is mostly distributed within 80 pixels, but it is unevenly distributed. PointRend extracted the most counters for errors between 1 and 7 pixels, which is consistent with the MPD index. Swin Transformer extracted the most counters from errors ranging from 7 to 15 pixels. This contradicts the MPD index results, and is inconsistent with the visual results.

Table 6. Evaluation Results of the Building Counter Accuracy Using HD Metric (Unit: Pixels).

Methods	Counts of Buildings	Maximum Error	Average Error	Mean Square Error
Mask R-CNN	1721	410.83	21.89	24.21
SwinTransformer	1873	412.95	19.87	22.86
PointRend	1798	441.23	26.84	38.88

Note: A lower HD value indicates higher accuracy.

During the calculation process, the HD index considered two curves as two-point sets, ignoring the geometric relationships of points on the same curve, and only calculated the distance between one pair of points on the two curves. As a result, the HD index was sensitive to unusual data points such as noise and outliers, which can cause deviation in distance calculation results, leading to larger errors. To summarize, while the HD index could effectively measure the similarity of two point sets in certain situations, it was insensitive to contour complexity and global distance. It could not accurately assess the precision of contour extraction.

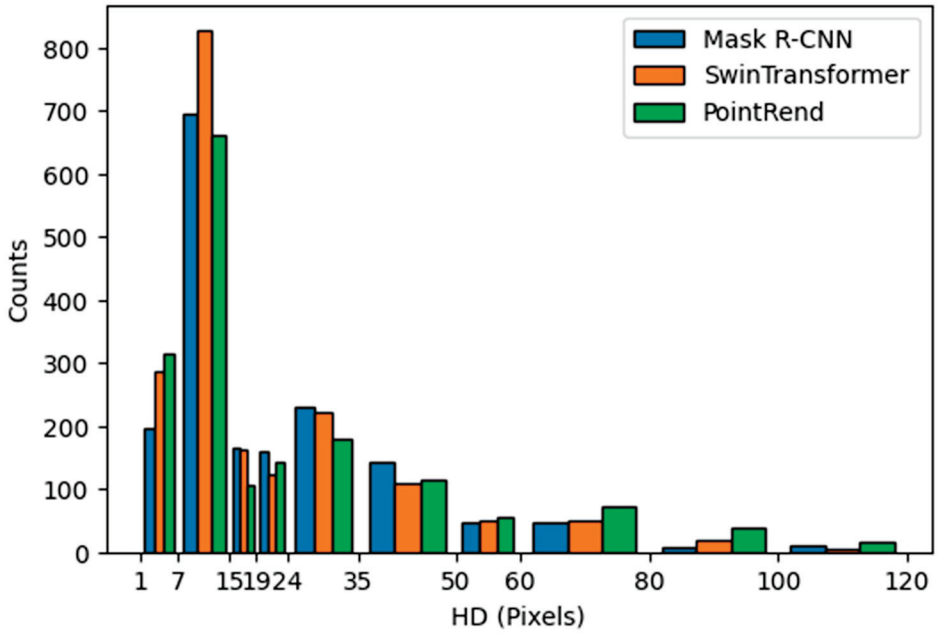


Figure 16. HD distribution histogram.

4.4. Accuracy Analysis of MPD Metrics

Table 7 displays the accuracy evaluation results of many building roof counters extracted by the Mask R-CNN, Swin Transformer, and PointRend. Mask R-CNN, Swin Transformer, and PointRend extracted 1721, 1873, and 1798 buildings out of 2619 building counters, respectively. Swin Transformer extracted the greatest number of buildings. The PointRend algorithm had the lowest mean error of the three algorithms, with an average error of 9.01 pixels. At the same time, Figure 17 shows that the errors were primarily distributed within 50 pixels, with only a few building outlines having errors greater than 50 pixels. The larger errors were caused by the fact that the building outlines were not extracted.

Table 7. Evaluation Results of Building Counter Accuracy Using MPD Metric (Unit: Pixels).

Methods	Counts of Buildings	Maximum Error	Average Error	Mean Square Error
Mask R-CNN	1721	287.321	11.87	17.02
SwinTransformer	1873	152.73	10.14	8.45
PointRend	1798	92.706	9.01	8.06

Note: A lower MPD value indicates higher accuracy.

Figure 16 also shows that the errors of the three algorithms were concentrated within 15 pixels. When using the extracted results for mapping, the allowed error should be within 15 pixels according to the scale. Because iterative subdivision improves edge accuracy, PointRend had the highest number of extractions within seven pixels of error, consistent with visual effects. The indication shows that: (1) In most cases, the extraction results of the PointRend algorithm were relatively stable, but in a few exceptional cases, the extraction results of the PointRend algorithm were poor. As a result, while PointRend did not extract the greatest number of buildings, it did extract the greatest number that met the accuracy requirements. (2) The proposed method could calculate the mapping accuracy of all tested buildings effectively (by comparing the AI-extracted building roof counters with their corresponding reference counters and using the mapping accuracy evaluation method proposed in this paper to calculate the accurate mapping accuracy of each tested building). As a result of the proposed mapping accuracy evaluation method's stability and high reliability, we demonstrated its capability.

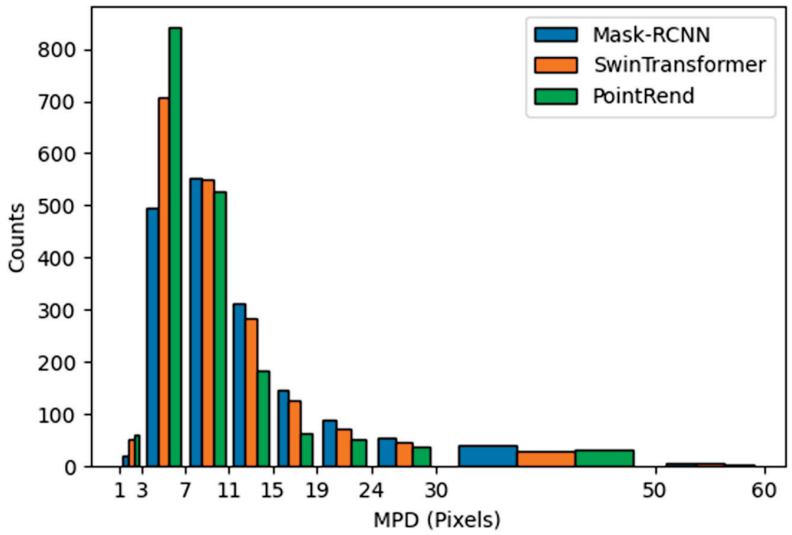


Figure 17. MPD distribution histogram.

5. Conclusions

With the rapid development of artificial intelligence, automatic mapping has become a possibility. In this study, we proposed an inflection point matching algorithm for vector polygons to match the inflection points of extracted building contours and reference contours. The definition and judgment conditions of edge inflection points were given. Aiming to address the problems that exist in the accuracy evaluation methods based on the IoU, we designed the mean inflection point distance (MPD), which can automatically calculate any two polygon contours (many-to-one and one-to-many inflection points between the extracted contour and the reference contour, with the extracted contour having more inflection points and more edges). The IoU could only calculate the percentage, not the exact error value, but the MPD can calculate the exact error value. The MPD value can be used to determine the scale suitable for mapping. Our experiments show that the MPD is more sensitive to the boundary accuracy than IoU and Boundary IoU, has a higher differentiation, and can obtain an accurate error value. The MPD has a better reference value for the accuracy requirements of mapping according to different scales. The relevant research in this paper provides a good reference for carrying out this accuracy evaluation method of artificial intelligence automated mapping. Our evaluation metrics begin with the regularization of the extracted contours, which affects the accuracy of the vector contours. As our next step, we will study the contour regularization technique to reduce the error impact of regularization.

Author Contributions: X.X. proposed the concept, provided data and data annotation, designed the method, and guided the entire research. A.L. conceived the foundation; D.Y. and Y.L. conceived and designed the research; D.Y. and J.L. wrote the code, processed the data and performed the experiments; D.Y., J.L. and Y.X. wrote the manuscript and reviewed and edited. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 61501470 and the Natural Science BasicResearch Plan in Shaanxi Province of China (Program No. 2021JQ-374).

Data Availability Statement: https://www2.isprs.org/commissions/comm2/wg4/benchmark/data-request-form/, accessed on 12 January 2023.

Acknowledgments: The authors thank ISPRS for providing the open-access and free aerial image dataset. The authors would also like to thank the anonymous reviewers and the editors for their insightful comments and helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Moser, G.; Serpico, S.B.; Benediktsson, J.A. Land-Cover Mapping by Markov Modeling of Spatial–Contextual Information in Very-High-Resolution Remote Sensing Images. *Proc. IEEE* **2013**, *101*, 631–651. [CrossRef]
2. Friedl, M.A.; McIver, D.K.; Hodges, J.C.F.; Zhang, X.Y.; Muchoney, D.; Strahler, A.H.; Woodcock, C.E.; Gopal, S.; Schneider, A.; Cooper, A.; et al. Global Land Cover Mapping from MODIS: Algorithms and Early Results. *Remote Sens. Environ.* **2002**, *83*, 287–302. [CrossRef]
3. Maus, V.; Camara, G.; Cartaxo, R.; Sanchez, A.; Ramos, F.M.; de Queiroz, G.R. A Time-Weighted Dynamic Time Warping Method for Land-Use and Land-Cover Mapping. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 3729–3739. [CrossRef]
4. Longbotham, N.; Chaapel, C.; Bleiler, L.; Padwick, C.; Emery, W.J.; Pacifici, F. Very High Resolution Multiangle Urban Classification Analysis. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 1155–1170. [CrossRef]
5. Li, X.; Xu, F.; Xia, R.; Li, T.; Chen, Z.; Wang, X.; Xu, Z.; Lyu, X. Encoding Contextual Information by Interlacing Transformer and Convolution for Remote Sensing Imagery Semantic Segmentation. *Remote Sens.* **2022**, *14*, 4065. [CrossRef]
6. Fritsch, J.; Kuhnl, T.; Geiger, A. A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013; pp. 1693–1700.
7. Zhang, H.; Liao, Y.; Yang, H.; Yang, G.; Zhang, L. A Local–Global Dual-Stream Network for Building Extraction from Very-High-Resolution Remote Sensing Images. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1269–1283. [CrossRef]
8. Cheng, G.; Wang, Y.; Xu, S.; Wang, H.; Xiang, S.; Pan, C. Automatic Road Detection and Centerline Extraction via Cascaded End-to-End Convolutional Neural Network. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3322–3337. [CrossRef]
9. Li, W.; He, C.; Fang, J.; Zheng, J.; Fu, H.; Yu, L. Semantic Segmentation-Based Building Footprint Extraction Using Very High-Resolution Satellite Images and Multi-Source GIS Data. *Remote Sens.* **2019**, *11*, 403. [CrossRef]
10. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
11. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
12. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2015; Volume 9351, pp. 234–241. ISBN 978-3-319-24573-7.
13. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*; Stoyanov, D., Taylor, Z., Carneiro, G., Syeda-Mahmood, T., Martel, A., Maier-Hein, L., Tavares, J.M.R.S., Bradley, A., Papa, J.P., Belagiannis, V., et al., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11045, pp. 3–11. ISBN 978-3-030-00888-8.
14. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef]
15. Dai, J.; He, K.; Sun, J. Instance-Aware Semantic Segmentation via Multi-Task Network Cascades. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3150–3158.
16. Luo, M.; Ji, S.; Wei, S. A Diverse Large-Scale Building Dataset and a Novel Plug-and-Play Domain Generalization Method for Building Extraction. *arXiv* **2022**, arXiv:2208.10004.
17. Ji, S.; Wei, S.; Lu, M. Fully Convolutional Networks for Multisource Building Extraction from an Open Aerial and Satellite Imagery Data Set. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 574–586. [CrossRef]
18. Zhu, Q.; Liao, C.; Hu, H.; Mei, X.; Li, H. MAP-Net: Multiple Attending Path Neural Network for Building Footprint Extraction From Remote Sensed Imagery. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 6169–6181. [CrossRef]
19. Wang, L.; Fang, S.; Meng, X.; Li, R. Building Extraction with Vision Transformer. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [CrossRef]
20. Jin, Y.; Xu, W.; Zhang, C.; Luo, X.; Jia, H. Boundary-Aware Refined Network for Automatic Building Extraction in Very High-Resolution Urban Aerial Images. *Remote Sens.* **2021**, *13*, 692. [CrossRef]
21. Fang, F.; Wu, K.; Liu, Y.; Li, S.; Wan, B.; Chen, Y.; Zheng, D. A Coarse-to-Fine Contour Optimization Network for Extracting Building Instances from High-Resolution Remote Sensing Imagery. *Remote Sens.* **2021**, *13*, 3814. [CrossRef]
22. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Zitnick, C.L. *Microsoft COCO: Common Objects in Context*; Springer International Publishing: Berlin/Heidelberg, Germany, 2014.
23. Rezatofighi, H.; Tsoi, N.; Gwak, J.Y.; Sadeghian, A.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
24. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 12993–13000. [CrossRef]

25. Cheng, B.; Girshick, R.; Dollár, P.; Berg, A.C.; Kirillov, A. Boundary IoU: Improving Object-Centric Image Segmentation Evaluation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
26. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
27. Heimann, T.; Van Ginneken, B.; Styner, M.A.; Arzhaeva, Y.; Aurich, V.; Bauer, C.; Beck, A.; Becker, C.; Beichel, R.; Bekes, G.; et al. Comparison and Evaluation of Methods for Liver Segmentation From CT Datasets. *IEEE Trans. Med. Imaging* **2009**, *28*, 1251–1265. [CrossRef] [PubMed]
28. Zhu, Y.; Huang, B.; Gao, J.; Huang, E.; Chen, H. Adaptive Polygon Generation Algorithm for Automatic Building Extraction. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
29. Wu, Y.; Xu, L.; Chen, Y.; Wong, A.; Clausi, D.A. TAL: Topography-Aware Multi-Resolution Fusion Learning for Enhanced Building Footprint Extraction. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
30. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Čehovin, L.; Vojír, T.; Häger, G.; Lukežič, A.; Fernández, G.; et al. The Visual Object Tracking VOT2016 Challenge Results. In *Lecture Notes in Computer Science, Proceedings of the Computer Vision—ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–16 October 2016*; Hua, G., Jégou, H., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 777–823.
31. Baltsavias, E.P. Object Extraction and Revision by Image Analysis Using Existing Geodata and Knowledge: Current Status and Steps towards Operational Systems. *ISPRS J. Photogramm. Remote Sens.* **2004**, *58*, 129–151. [CrossRef]
32. Lowe, D.G. Object Recognition from Local Scale-Invariant Features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157.
33. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
34. Di, Y.; Shao, Y.; Chen, L.K. Real-Time Wave Mitigation for Water-Air OWC Systems Via Beam Tracking. *IEEE Photonics Technol. Lett.* **2021**, *34*, 47–50. [CrossRef]
35. Leal-Taixé, L.; Milan, A.; Reid, I.; Roth, S.; Schindler, K. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv* **2015**, arXiv:1504.01942.
36. Automated Segmentation of Colorectal Tumor in 3D MRI Using 3D Multiscale Densely Connected Convolutional Neural Network. Available online: https://www.hindawi.com/journals/jhe/2019/1075434/ (accessed on 20 February 2023).
37. Hung, W.-L.; Yang, M.-S. Similarity Measures of Intuitionistic Fuzzy Sets Based on Hausdorff Distance. *Pattern Recognit. Lett.* **2004**, *25*, 1603–1611. [CrossRef]
38. Rote, G. Computing the Minimum Hausdorff Distance between Two Point Sets on a Line under Translation. *Inf. Process. Lett.* **1991**, *38*, 123–127. [CrossRef]
39. Suzuki, S.; Be, K. Topological Structural Analysis of Digitized Binary Images by Border Following. *Comput. Vis. Graph. Image Process.* **1985**, *30*, 32–46. [CrossRef]
40. Douglas, D.H.; Peucker, T.K. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature. In *Classics in Cartography*; Dodge, M., Ed.; Wiley: Hoboken, NJ, USA, 2011; pp. 15–28. ISBN 978-0-470-68174-9.
41. Petrakis, E.; Diplaros, A.; Milios, E. Matching and Retrieval of Distorted and Occluded Shapes Using Dynamic Programming. *Pattern Anal. Mach. Intell. IEEE Trans.* **2002**, *24*, 1501–1516. [CrossRef]
42. Kirillov, A.; Wu, Y.; He, K.; Girshick, R. PointRend: Image Segmentation as Rendering. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 9796–9805.
43. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 9992–10002.
44. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
45. Rottensteiner, F.; Sohn, G.; Gerke, M.; Wegner, J.D.; Breitkopf, U.; Jung, J. Results of the ISPRS Benchmark on Urban Object Detection and 3D Building Reconstruction. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 256–271. [CrossRef]
46. Jozdani, S.; Chen, D. On the Versatility of Popular and Recently Proposed Supervised Evaluation Metrics for Segmentation Quality of Remotely Sensed Images: An Experimental Case Study of Building Extraction. *ISPRS J. Photogramm. Remote Sens.* **2020**, *160*, 275–290. [CrossRef]
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

remote sensing

MDPI

Article

Research on SUnet Winter Wheat Identification Method Based on GF-2

Ke Zhou [1,2], Zhengyan Zhang [1,2], Le Liu [1,2], Ru Miao [1,2,*], Yang Yang [1,2], Tongcan Ren [1,2] and Ming Yue [1,2]

[1] School of Computer and Information Engineering, Henan University, Kaifeng 475004, China
[2] Henan Province Engineering Research Center of Spatial Information Processing, Henan University, Kaifeng 475004, China
* Correspondence: mr1015@henu.edu.cn

Abstract: Introduction: Winter wheat plays a crucial role in ensuring food security and sustainable agriculture. Accurate identification and recognition of winter wheat in remote sensing images are essential for monitoring crop growth and yield estimation. In recent years, attention-based convolutional neural networks have shown promising results in various image recognition tasks. Therefore, this study aims to explore the application of attention-based convolutional neural networks for winter wheat identification on GF-2 high-resolution images and propose improvements to enhance recognition accuracy. Method: This study built a multi-band winter wheat sample dataset based on GF-2 images. In order to highlight the characteristics of winter wheat, this study added two bands, NDVI and $NDVI_{increase}$, to the dataset and proposed a SUNet network model. In this study, the batch normalization layer was added to the basic structure of the UNet convolutional network to speed up network convergence and improve accuracy. In the jump phase, shuffle attention was added to the shallow features extracted from the coding structure for feature optimization and spliced with the deep features extracted by upsampling. The SUNet made the network pay more attention to the important features to improve winter wheat recognition accuracy. In order to overcome the sample imbalance problem, this study used the focus loss function instead of the traditional cross-entropy loss function. Result: The experimental data show that its mean intersection over union, overall classification accuracy, recall, F1 score and kappa coefficient are 0.9514, 0.9781, 0.9707, 0.9663 and 0.9501, respectively. The results of these evaluation indicators are better than those of other comparison methods. Compared with the UNet, the evaluation indicators have increased by 0.0253, 0.0118, 0.021, 0.0185, and 0.0272, respectively. Conclusion: The SUNet network can effectively improve winter wheat recognition accuracy in multi-band GF-2 images. Furthermore, with the support of a cloud platform, it can provide data guarantee and computing support for winter wheat information extraction.

Keywords: winter wheat identification; SUNet; remote sensing image

1. Introduction

As one of the three major grain crops in China, wheat occupies an important position in the daily life of the people and the national economy [1]. Accurately identifying winter wheat from high-resolution images requires remote sensing technology and convolutional neural networks [2,3]. This is paramount to winter wheat spatial distribution accuracy and ensuring grain yield [4].

Convolutional neural networks (CNN) can automatically extract abstract features and reduce the number of parameters in the neural network during the deep propagation process [5]. It is widely used in image recognition [6–10], object detection [11–13], semantic segmentation [14–18], remote sensing scene classification [19–21], image denoising [22–24] and other fields. The extensive application of convolutional neural networks in computer vision signifies that their techniques and methods can be effectively employed in remote

sensing [25,26] image automatic interpretation. The use of image semantic segmentation has become an important technology in order to extract the data regarding crops from high-resolution images. For example, Zhang et al. [27] constructed a hybrid structure convolutional neural network (HSCNN) with two different depth components based on CNN which was used to extract winter wheat planting information in the study area. Compared with SegNet [28] and DeepLab [29], their method improves accuracy by 0.147 and 0.059, respectively, and achieves good results. Chen [30] constructed a multi-scale feature convolution neural network (MSFCNN) model to extract winter wheat planting information in the study area. Furthermore, Wang et al. [31] proposed a method to improve the accuracy of image edge pixel extraction, thereby improving the classification accuracy of winter wheat. This approach used a partially connected conditional random field model (PCCRF) to refine RefineNet [32] classification results. The results showed that this method effectively extracted winter wheat spatial distribution information. In addition, Teimouri et al. [33] developed and implemented a lightweight network structure. This network structure is based on deep learning and combines a fully convolutional network and a convolutional long short-term memory network (ConvLSTM). The authors applied this network structure to the spatial information extraction of 14 crops in their study area. The average pixel-based accuracy and Intersection over Union obtained from the proposed network were 86% and 0.64, respectively, and achieved good results. In remote sensing image semantic segmentation, the encoder–decoder structure similar to that of the UNet networks to stitch low-level and high-level features achieved better results.

However, the current research primarily relies on Sentinel and Landsat data, with limited utilization of Chinese GF data. This is mainly due to the relative difficulty in accessing Chinese GF data. Chinese GF data is typically controlled and managed by the government or relevant organizations, requiring specific application procedures or appropriate permissions to access such data. The spatial resolution of GF-2 data in China can reach the sub-meter level. The GF-2 PMS data can provide optical images at a 0.8 m resolution, which greatly exceeds Sentinel (10 m) and Landsat (30 m) capabilities, thus providing information on winter wheat, e.g., texture and field shape. However, there are few public datasets of GF-2 winter wheat for training high-resolution images, and traditional machine learning methods still need to extract features artificially. As a result, only shallow features can be learned when training on high-resolution images, which affects the final target recognition accuracy. Furthermore, due to the different planting area sizes of winter wheat, there are different patch sizes in remote sensing images, which causes unbalanced samples in the dataset. Determining ways to construct a high-resolution remote sensing image dataset of winter wheat and use a convolutional neural network to accurately identify winter wheat information on high-resolution images are also problems to be solved.

This paper proposes a SUNet (Shuffle Attention UNet) network model combined with an attention mechanism. The primary purpose is to solve the problem of unbalanced sample datasets and high-resolution winter wheat extraction tasks. In order to make the neural network pay more attention to important features, a batch normalization layer is added to improve the target recognition accuracy after layer convolution. In the decoding stage, the underlying features of the encoding stage are optimized to strengthen the network model's area of interest. Then, this paper uses the focus loss function to calculate the loss value. This can alleviate the problem of sample imbalance. Thus, the classification accuracy of winter wheat can be improved. In addition, the data enhancement strategy can be used to expand the winter wheat dataset during training. This makes the network model more robust, and it also avoids overfitting.

In summary, the objectives of this study are as follows:

(1) Construct a high-resolution winter wheat public dataset based on the China GF-2 satellite. The dataset contains six bands of RGB, near-infrared, NDVI and $NDVI_{increase}$, and has rich image samples and labeling information.

(2) Propose the SUNET network model, which introduces the Batch normalization layer and the Shuffle Attention mechanism. The results of the comparison test and the ablation experiment show that the generalization ability and classification accuracy of the SUNET model have been improved.

2. Methodology

2.1. Research Area and Data Source

2.1.1. Research Area

This paper selected Kaifeng City, Henan Province as the research area. Kaifeng City is located in the east of Henan Province, between 34°11′43″–35°11′43″N and 113°51′51″–115°15′42″E, as shown in Figure 1. The east–west length is about 126 km, the north–south width is about 92 km, and the total area is 6266 square kilometers. Kaifeng City is located in the eastern part of the alluvial plain in the middle and lower reaches of the Yellow River, with a 69–78 m altitude and flat terrain. The climate is distinct in four seasons and belongs to a temperate monsoon climate. The general climate characteristics are high temperature and rainy weather in summer and cold and dry weather in winter. The annual average frost-free period is 221 days, the average yearly temperature is 14.52 °C, the annual average precipitation is 627.5 mm, and the precipitation is mainly concentrated in July and August in summer. Kaifeng City is rich in groundwater resources, and the natural ecological environment is suitable for the growth of various crops.

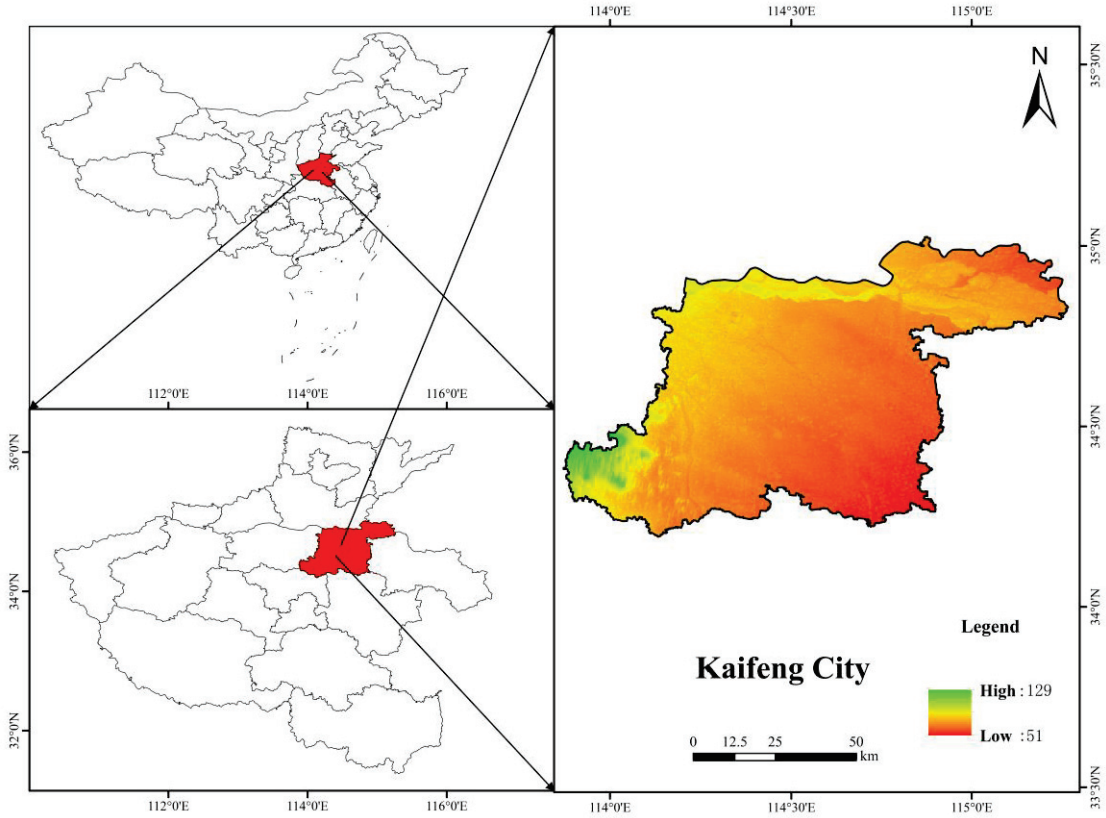


Figure 1. Study area location map.

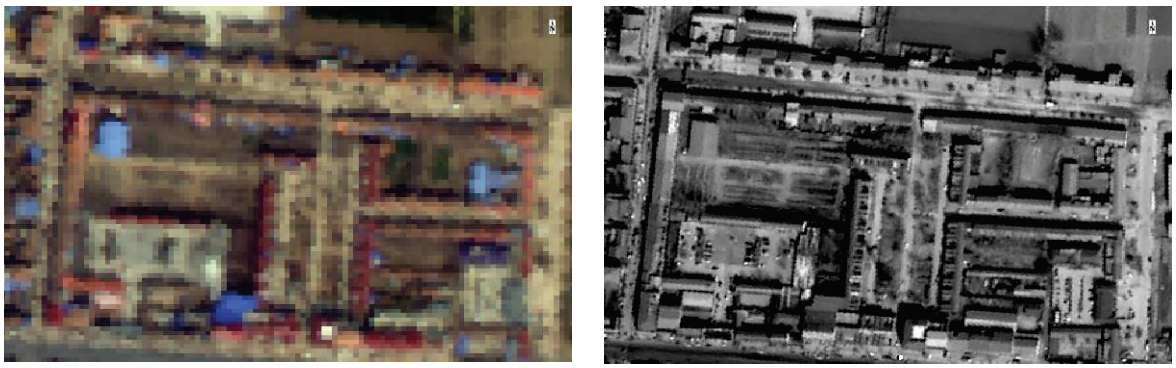
2.1.2. Data Source

GF-2 satellite is the first civil optical remote sensing satellite with a resolution better than 1 m independently developed by China. It was successfully launched on 19 August 2014. It was first imaged and downloaded on 21 August 2014. The detailed parameters of GF-2 are shown in Table 1. It has the characteristics of high-speed positioning, sub-meter spatial resolution and fast maneuvering side swing ability, which effectively improves the satellite's comprehensive observation ability. The image sample is shown in Figure 2.

At present, GF-2 satellite plays an important role in land and resource survey, urban fine management, mineral resource monitoring, crop growth monitoring and yield estimation, forest resource survey, desertification monitoring, geological disaster monitoring, flood monitoring and post-disaster reconstruction.

Table 1. GF-2-related parameter information.

Sensor Type	Spatial Resolution	Band Number	Spectral Range	Width	Remark
Panchromatic	1 m	1	0.45~0.90 μm	45 km	
		2	0.45~0.52 μm		Blue
		3	0.52~0.59 μm		Green
Multispectral	4 m	4	0.63~0.69 μm		Red
		5	0.77~0.89 μm		Near infrared



(**a**) (**b**)

Figure 2. GF-2 data sample. (**a**) Multispectral bands; (**b**) panchromatic band.

The experimental data used the GF-2 image data of the western region of Lankao County, Kaifeng City on 5 March 2020. During this period, winter wheat is in the rising and jointing stage, which is more obvious than other features [34]. Wheat seedlings are green in remote sensing images. In addition, GF-2 image data from 28 May in the same year and the same region were selected for comparison to improve the sample production accuracy of winter wheat. At this time, winter wheat was in the milk-filling stage, and wheat seedlings were brown in true color images. The two-image data selected in the experiment have less cloud content and higher clarity.

2.2. Technical Process

GF-2 image data were selected to extract winter wheat spatial distribution information in the study area in 2020 using a convolution neural network. At the same time, an improved UNet neural network was applied to improve winter wheat recognition accuracy. The process steps are as follows, as shown in Figure 3: Step 1—data preprocessing operations are performed on the GF-2 image, including radiometric calibration, atmospheric correction, orthorectification and image fusion. Then, the processed images are marked to make a winter wheat sample dataset; Step 2—the improved UNet convolutional neural network model is constructed; Step 3—the research on the accuracy of winter wheat recognition is realized using the improved convolutional neural network on the AI studio platform of Baidu; Step 4—the experimental results are analyzed by comparing the experiment and evaluation indicators.

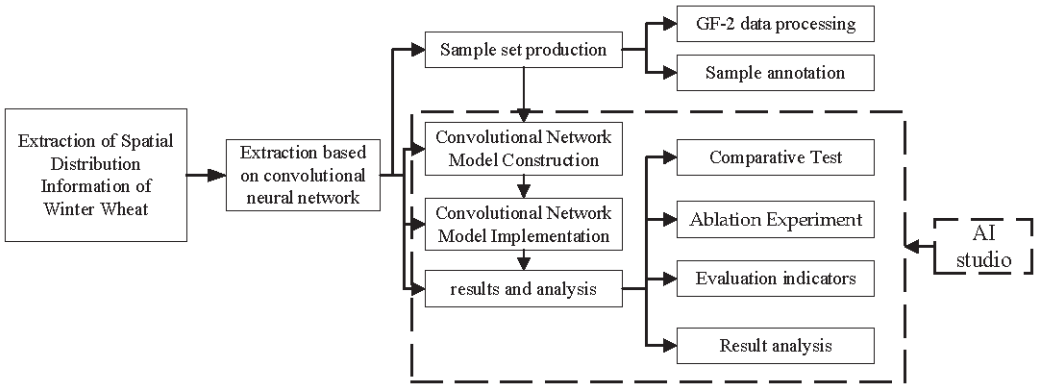


Figure 3. Technical flow chart.

2.3. SUNet Network Construction

UNet is a pixel-level image segmentation network that performs well on small sample datasets, and its function is to identify and label each pixel of the image [35]. The UNet network adopts a U-shaped symmetrical structure. The compression channel (encoder) and expansion channel (decoder) of convolution and pooling operation are on their left and right sides. The training set was extracted layer by layer through the encoding–decoding structure. The function of the encoder part of the network is to perform multi-scale feature recognition and extraction on the image matrix through four pooling operations. Feature fusion at different scales improves the classification accuracy of images. This paper extracted the high-resolution winter wheat data, improved the UNet model, and proposed the SUNet network model. A batch normalization layer (BN) was added after each convolution layer of the UNet network to improve the training speed and convergence speed and prevent overfitting. In this paper, the Shuffle Attention was used instead of traditional direct connection in the feature splicing stage of the UNet network decoding structure. Including the Shuffle Attention mechanism can strengthen the region of interest of the model and achieve selective and full feature extraction. In this way, the network can accurately focus on all the relevant elements of the input, weaken the random noise caused by the network misjudgment, and simultaneously better reduce the computational overhead.

The model structure of SUNet is shown in Figure 4. The model is divided into 5 parts: input, encoder, decoder, classification and output. The model's input is the size of the remote sensing image sample c $\times$ h $\times$ w, where h and w are the height and width of the sample data, set to 512 $\times$ 512, and c is the number of input channels. In this paper, multi-band data were used to extract winter wheat. In addition to the four bands of the GF-2 image, two features of NDVI and $NDVI_{increase}$ were added, so c was set to 6. Each layer of the encoder structure contains two 3 $\times$ 3 convolutions, each followed by a BN layer and a ReLU layer. A 2 $\times$ 2 maximum pool downsampling was used to obtain shallow features. In the decoder part, bilinear interpolation was used for upsampling, and Shuffle Attention optimized the shallow features obtained in the encoder stage through jump links. Then, the image information was restored by splicing and fusion with the deep features extracted by upsampling. The classifier uses a 1 $\times$ 1 convolution to output the multi-channel features, and the output result contains n channels. Since this paper dealt with binary classification, specifically the categorization of remote sensing images into winter wheat and non-winter wheat classes, the value of n was set to 2. Finally, the softmax function calculated the mapping of each pixel value of the classification feature graph to the real number [0prime1]. The output image is a single-channel image of the same size as the input image. Each pixel value represents the category to which it belongs, 0 for non-winter wheat and 1 for winter wheat.

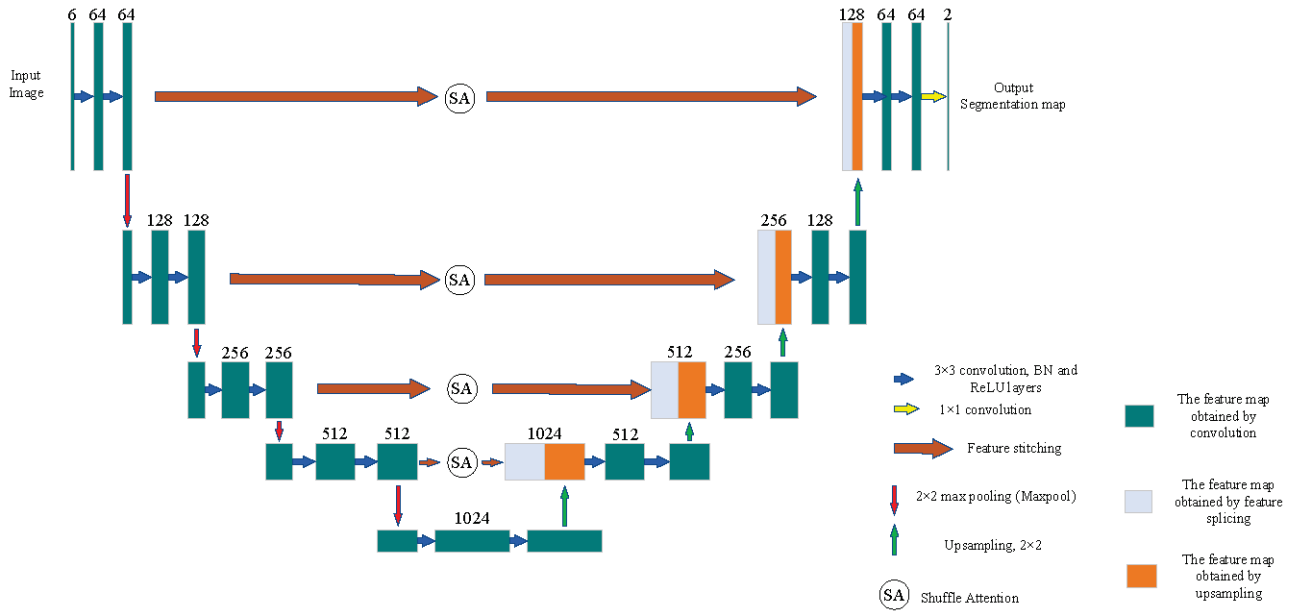


Figure 4. SUNet network structure diagram.

2.3.1. Batch Normalization

Batch normalization is a neural network optimization method proposed by Ioffe et al. [36]. Normalization can make scattered data regular, so the neural network can better learn the rules between the data. BN processing calculates the variance and mean of each batch of data. In the forwarding transmission of each batch of data, the input data of each layer is normalized by the recorded mean and variance. This ensures that the input distribution at each network layer is the same. It can also effectively improve the convergence speed of the neural network, prevent the gradient from disappearing and alleviate the overfitting phenomenon. This paper improved the UNet model by adding the BN layer after the convolution layer of each network layer. This can improve the network's generalization ability, standardize the network layer's data input, effectively prevent the disappearance of the gradient, accelerate the convergence speed and alleviate overfitting to improve winter wheat classification accuracy. The basic structure of the convolution network added to the BN layer is shown in Figure 5.

2.3.2. Shuffle Attention (SA)

The SA mechanism was proposed by Zhang et al. [37] and others from Nanjing University. Firstly, the module uses group convolution to group different input feature maps to reduce computation. Then, channel segmentation is used for each data group. One part is used for spatial attention calculation, and the other participates in channel attention calculation. Concat fuses the feature information of the same group, and finally the sub-feature images obtained through different channel attention mechanisms are reconstructed by channel shuffle operation. Spatial attention uses group normalization (GroupNorm, GN) [38] to obtain information on the spatial dimension. Channel attention uses this global average pooling [39] operation to generate channel-related statistics. Then, the parameters are used to scale and translate channel vectors and generate channel and spatial feature representations. Without increasing the model's size, the UNet network is improved in this paper. In the jump phase, the Shuffle Attention module is added to optimize the features of the coding phase. Then, it is spliced with the features obtained by upsampling to make the neural network pay more attention to the useful features and restrain the useless parts in the decoding process. Thus, the convolution network improves the recognition accuracy of winter wheat in high-resolution images.

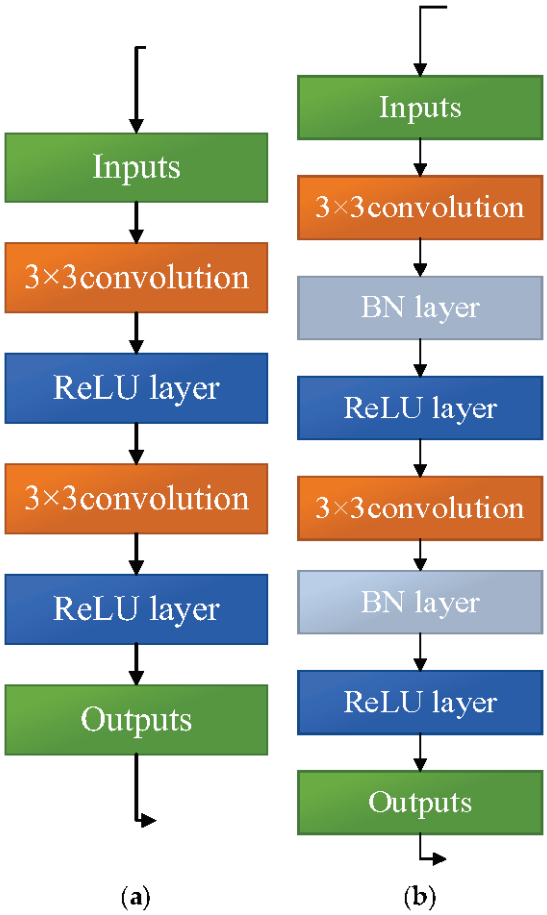


Figure 5. Structure comparison. (**a**) UNet original convolutional layer structure; (**b**) UNet convolutional layer structure with added BN layer.

2.4. Experimental Setup

2.4.1. Lab Environment

The programming language is Python. In addition, the NumPy library, GDAL library, PIL library and Matplotlib library are used to realize data processing and visualization. This paper implements data enhancement processing, construction, training, testing, and results in analyzing the SUNet neural network based on Baidu's open-source deep learning framework, PaddlePaddle, on the AI studio platform. The detailed experimental configuration environment is shown in Table 2.

Table 2. Experimental environment (AI studio) configuration information.

Surroundings	Name
Deep Learning Framework	PaddlePaddle-gpu 2.2.0
Programming language	Python 3.7
CPU	Intel® X®(R) Gold 6271C (2.6 Hz)
GPU	Tesla V100 (32 G)
RAM	32 G
Hard disk	100 G

2.4.2. Data Pre-Processing

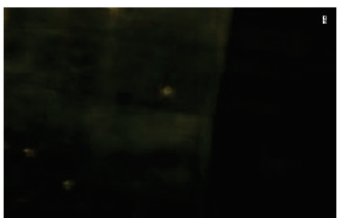
Due to the influences of factors such as satellite altitude, atmospheric conditions, and observation environment, remote sensing images often exhibit distortions and information errors [40]. Therefore, it is necessary to preprocess the remote sensing images before further analysis. In order to obtain more authentic and reliable remote sensing images, a series

of preprocessing operations were conducted on the selected high-resolution GF-2 satellite imagery using ENVI 5.3 software. These operations included radiometric calibration, atmospheric correction [41], orthorectification, and image fusion [42]. Through these preprocessing steps, distortions caused by sensor and environmental factors were mitigated, leading to improved accuracy and reliability of the images.

Additionally, to enhance the extraction effectiveness of winter wheat information and improve data visualization, the band combination and stretching techniques were applied to the remote sensing images. The band combination process fused data from different spectral bands, providing more comprehensive information, while preserving the 8-bit data to ensure data accuracy. The stretching process adjusted the brightness and contrast of the images using histogram equalization, resulting in enhanced visibility of details. These processing techniques enabled us to achieve better visual effects and laid a solid foundation for subsequent winter wheat patch extraction and analysis.

Deep learning requires a large number of data samples. Due to the diversity of remote sensing images and the continuous changes in the growth stages of winter wheat, there is a limited availability of publicly available datasets specifically related to winter wheat. Therefore, the high-resolution GF-2 images in this study were classified and annotated to create a dataset suitable for deep learning training. The images were labeled into two categories, winter wheat and non-winter wheat. The category features are shown in Table 3. The non-winter wheat in Table 3 refers to water bodies, buildings, roads, and other vegetation categories; the color changes in water bodies, buildings, roads, etc., in the front and rear images are small, and the changes in other vegetation are opposite to winter wheat and gradually turn green. From the pictures in Table 3, it can be seen that the color of winter wheat is green in the image of March, and brown in the image of May, with obvious changes and clear texture features. In order to obtain a more accurate winter wheat dataset, this study manually annotated the winter wheat on overlaid images from four different sources: (1) Training images, which were GF-2 satellite imagery from 5 March 2020; (2) Winter wheat mask data extracted using the random forest research method [34]; (3) GF-2 satellite imagery from the same area during 25 May 2020, when winter wheat underwent noticeable color changes compared with the training images; (4) Survey data from Lankao County. Based on the survey data, this paper's GF-2 satellite imagery from March was annotated by combining the winter wheat mask regions and the variations of winter wheat in different growth stages in the imagery.

Table 3. Features of remote sensing image categories.

Tag Category	Tag Value	March Features	May Features
Non-winter wheat	0		
Winter wheat	1		

2.4.3. Loss Function

Because the planting range of winter wheat is relatively concentrated, and some areas are scattered, the distribution of categories are inevitably not balanced in the sample dataset.

Some samples have a large proportion of winter wheat, while others have more non-winter wheat. The traditional two-category cross-entropy loss function calculation formula is shown in Equation (1),

$$\mathrm{CE}(\mathrm{r},\mathrm{p}) = -\mathrm{r}\log\mathrm{p} - (1-\mathrm{r})\log(1-\mathrm{p}) = \begin{cases} -\log\mathrm{p}, \mathrm{r}=1 \\ -\log(1-\mathrm{r}), \mathrm{r}=0 \end{cases} \quad (1)$$

where r represents the true sample value and p represents the predicted class prediction probability during model training. According to the Equation (1), for positive samples, the higher the probability, the smaller the loss value; for negative samples, the lower the probability, the lower the loss value. This aligns with the principle that the more accurate the prediction, the lower the loss. However, when the sample category distribution is uneven, the loss value produced by many class samples covers up the loss value produced by a small number of class samples. This causes the gradient decline direction of the loss function to be inconsistent with the direction this paper expects to achieve. Based on this problem, Lin et al. [43] proposed an improved focus loss function based on the cross-entropy loss function (Focal Loss). The formula is shown in Equation (2).

$$\mathrm{FL} = \begin{cases} -\alpha(1-\mathrm{p})^{\gamma}\log\mathrm{p}, \mathrm{y}=1 \\ -(1-\alpha)\mathrm{p}^{\gamma}\log(1-\mathrm{p}), \mathrm{y}=0 \end{cases} \quad (2)$$

where α and γ are adjustable variables. It can be seen from the formula that the weight of the loss value of the classification result is increased compared with the two-classification cross-entropy loss function. As a result, the loss weight of the positive sample is low, and the weight loss of the negative sample is high, which can effectively solve the problem of sample imbalance. Therefore, this paper chose Focal Loss as the loss function to calculate the loss value of the classification results.

2.4.4. Training Process

The training parameters used in this paper are shown in Table 4. After many experiments, the epoch value was set to 200, the batch size was set to 4, the learning rate was set to 0.0001, and the cosine annealing decay attenuated the learning rate. In order to update the parameters more accurately, it is necessary to optimize the network parameters. The Adam optimization algorithm performs excellently and is widely used in neural network optimization. Therefore, this paper used the Adam optimization algorithm to optimize the model.

Table 4. Details of training parameters.

Hyperparameter Name	Parameter Value
epoch	200
Batch size	4
Initial learning rate	0.0001
Learning rate decay method	Cosine annealing decay
Optimizer	Adam

This paper used the parameters in Table 4 to train and test the built network model. The training process is as follows, and the process is shown in Figure 6:

(1) The super parameter initialization of the network model training process is determined.
(2) The training data are input into the network model for forwarding calculation. The features are extracted by the convolution layer, pooling layer and deconvolution layer hidden in the network coding and decoding part. Finally, all the sample pixels are classified in the classification layer to obtain a set of predictive values xp.
(3) The focal loss function is used to calculate the error between the predicted value xp and the true value. If the error meets the target requirement, the training is completed. Otherwise, the training is continued.

(4) The loss value is derived by the Adam optimizer, and the parameters are back-propagated to realize the parameter update of the SUNet network model, thus reducing the loss value.

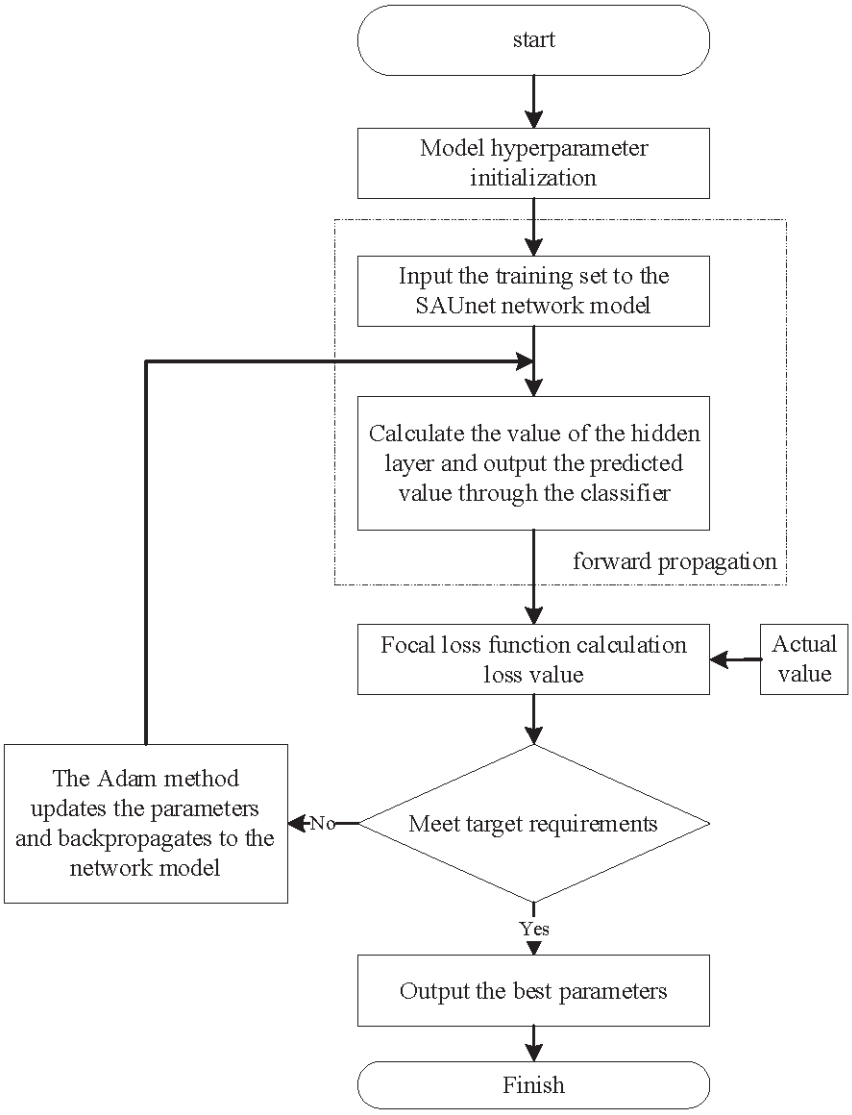


Figure 6. Neural network training process.

2.4.5. Evaluation Metrics

This paper quantitatively evaluated the classification accuracy of the network model used in this paper by employing metrics commonly utilized in semantic segmentation. These metrics include mean intersection ratio (mIou), overall classification accuracy (OA), precision (precision), recall (recall), F1 score, and kappa coefficient.

The calculation formula of the average intersection and union ratio is shown in Equation (3). It represents the ratio of the intersection and union of the actual and predicted values.

$$\mathrm{mIou} = \frac{1}{k+1} \sum_{i=0}^{k-1} \frac{P_{ij}}{\sum_{j=0}^{k-1} P_{ij} + \sum_{j=0}^{k-1} P_{ji} - P_{ii}}. \quad (3)$$

The formula for calculating the overall classification accuracy is shown in Equation (4). It represents the ratio of correctly classified pixels to the total number of pixels.

$$\mathrm{OA} = \frac{\sum_{i=0}^{k-1} P_{ii}}{\sum_{i=0}^{k-1}\sum_{j=0}^{k-1} P_{ij}}. \tag{4}$$

The formula for calculating the accuracy rate is Equation (5). It represents the ratio of the correctly classified pixels of winter wheat to the total number of predicted winter wheat pixels.

$$\mathrm{precision} = \sum_{i=0}^{k-1} \frac{P_{ii}}{\sum_{j=0}^{k-1} P_{ij}}. \tag{5}$$

The formula for the recall rate is shown in Equation (6). It represents the ratio of the number of correctly classified pixels of winter wheat to the total number of pixels of real winter wheat.

$$\mathrm{recall} = \sum_{j=0}^{k-1} \frac{P_{jj}}{\sum_{i=0}^{k-1} P_{ij}}. \tag{6}$$

The F1 score calculation formula is shown in Equation (7). It is a comprehensive evaluation index that can consider the accuracy of the classification model and the recall rate.

$$\mathrm{F1} = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}. \tag{7}$$

The kappa coefficient is used to measure the classification effect of winter wheat, and its calculation formula is shown in Equation (8),

$$\mathrm{kappa} = \frac{\mathrm{OA} - \frac{\sum_{x=0}^{k-1}\left[\sum_{i=1}^{k-1} P(i,x) * \sum_{j=1}^{k-1} P(x,j)\right]}{\left[\sum_{i=0}^{k-1}\sum_{j=0}^{k-1} P(i,j)\right]^2}}{1 - \frac{\sum_{x=0}^{k-1}\left[\sum_{i=1}^{k-1} P(i,x) * \sum_{j=1}^{k-1} P(x,j)\right]}{\left[\sum_{i=0}^{k-1}\sum_{j=0}^{k-1} P(i,j)\right]^2}}, \tag{8}$$

where P represents the calculated confusion matrix. When $i \neq j$, P_{ij} represents the number of pixels in which class I is mistakenly divided into class j, P_{ij} represents the number of pixels in class j misclassified into class I, P_{ij} represents the number of pixels correctly classified, and k represents the number of categories.

2.4.6. Experimental Description

This study conducted multiple sets of experiments for comparative analysis to verify the effectiveness and accuracy of winter wheat extraction in high-resolution remote sensing images. These experiments included popular semantic segmentation models, such as UNet, SegNet, U2-Net, and DeeplabV3+. Two network architectures were designed for ablation experiments to compare the impact of batch normalization (BN) layer and Shuffle Attention (SA) mechanism on performance improvement. The first one is Net-1, which includes the BN layer but does not have the SA mechanism. The second one is Net-2, which does not include the BN layer but has the SA mechanism.

3. Experimental Section and Results

3.1. Datasets

The authors produced the dataset used in this paper. This dataset has been published in Kaggle and linked in the data availability statement. This winter wheat dataset is a multi-band remote sensing image classification dataset. After analyzing the NDVI changes during the growth period of winter wheat, it was observed that the NDVI increase in winter wheat underwent significant variations compared with other land types. Moreover, both the NDVI values and the magnitude of $\mathrm{NDVI}_{\mathrm{increase}}$ during the growth stage directly

impacted the accuracy of winter wheat extraction. Therefore, first, this paper inverted the NDVI on the GF-2 image using the ENVI software. Second, the $NDVI_{increase}$ extracted by the method proposed by Wang et al. [44] was resampled according to the projection method, and the spatial resolution of GF-2 and data were normalized. Finally, two bands were added to the original band of the GF-2 image, namely NDVI and $NDVI_{increase}$ extracted from the Landsat 8 image. The $NDVI_{increase}$ formula [34] is shown in Equation (9). $NDVI_{min}$ is the minimum NDVI composite image of the image set from September to November in the early sowing period. $NDVI_{max}$ is the maximum NDVI composite image of the image set from December to March of the following year.

$$NDVI_{increase} = \frac{NDVI_{max} - NDVI_{min}}{|NDVI_{min}|}. \quad (9)$$

After the remote sensing image processing in Section 2.4.3 and the NDVI inversion and data normalization processing in the previous paragraph, this paper used the GDAL library in the Python language to read the raster image data and segmented the classified image and the labeled image in the same way. When the segmentation was completed, the labeled image was converted into a single-channel PNG format image. After the above process, a total of 648 groups of data samples was collected, and each image size was 512×512. These images were allocated in an 8:2 manner, in which 80% of the data was used for image training and 20% for classification accuracy verification. The winter wheat sample data is shown in Figure 7.

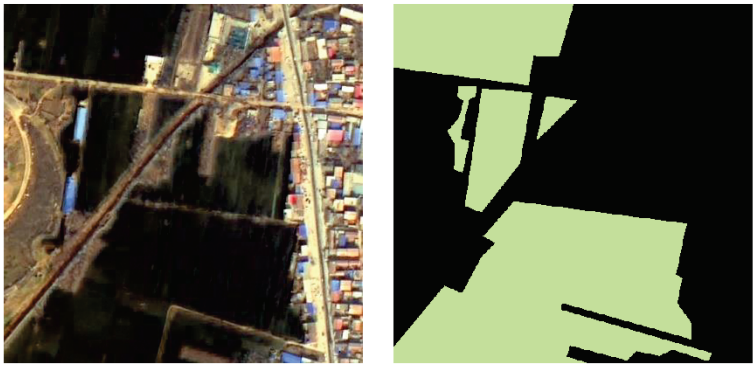


Figure 7. Sample of winter wheat (Green area is winter wheat, black area is non-winter wheat).

Data Augmentation

In deep learning, the network model is prone to overfitting when the sample data are limited. This study used a small and limited winter wheat sample dataset. To enhance the generalization ability of the network model during the training process and avoid overfitting, this paper used data augmentation techniques to expand the sample dataset without altering the labeling results.

Specifically, various data augmentation strategies were applied, including horizontal flipping, vertical flipping, 90° and 180° rotation, as well as diagonal flipping (diagonal mirroring), as shown in Figure 8. These operations generated a larger set of variant sample data while preserving the original sample's labeling results. These augmentation techniques aimed to increase the samples' diversity, enabling the network model to learn the winter wheat features under different variations, thus improving the winter wheat patch boundary recognition and extraction accuracy.

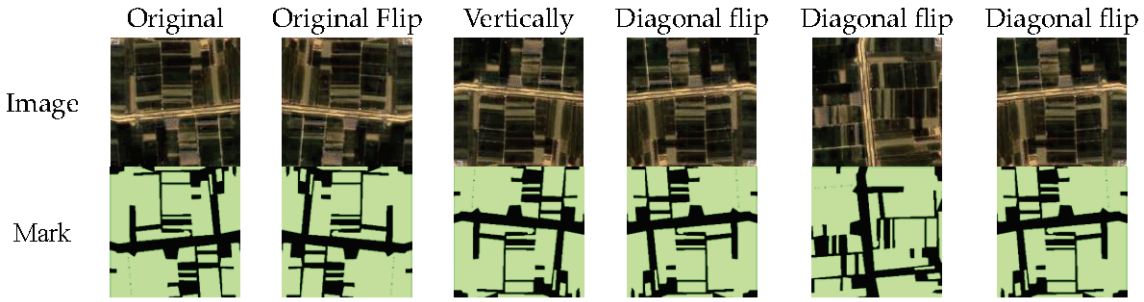


Figure 8. Data augmentation example.

3.2. Results

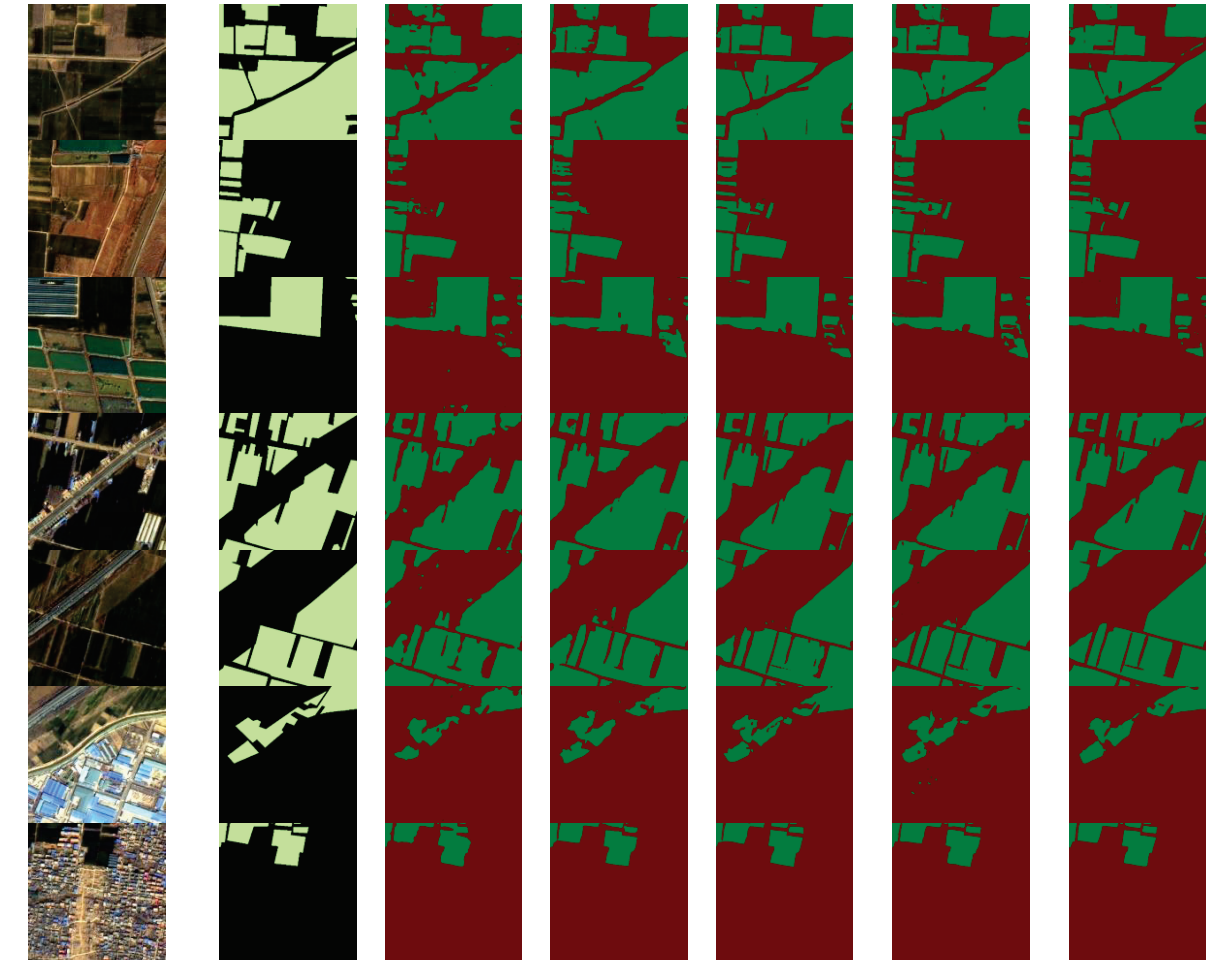
3.2.1. Comparative Test Result

The verified results are shown in Figure 9, and the evaluation indicators are shown in Table 5. Table 5 shows that after adding the BN layer and the Shuffle Attention mechanism to the UNet network structure, the indicators of the structure are significantly improved compared with the original network—MIou increased by 2.53%, OA increased by 1.18%, and the classification accuracy of winter wheat was improved by 1.59%. The classified recall rate of winter wheat increased by 2.1%, the F1 score by 1.85%, and the kappa coefficient by 2.72%. Table 5 also shows that compared with the classical image semantic segmentation networks SegNet, U^2-Net and Deeplabv3+, SUNet performs better in all indicators. Regarding the comparison of the size of the model, after adding the BN layer and the SA mechanism, the size of the model after SUNet training is only 0.1 m bigger than that of the UNet network. In addition, it is much smaller than the model size of SegNet, U^2-Net and Deeplabv3+.

Table 5. Evaluation index results of each network model.

Index \ Model	Unet	SegNet	U^2-Net	Deeplabv3+	SUNet
mIou	0.9261	0.9348	0.9437	0.9422	0.9514
OA	0.9663	0.9703	0.9746	0.9739	0.9781
precision	0.9460	0.9490	0.9596	0.9560	0.9619
recall	0.9497	0.9597	0.9619	0.9633	0.9707
F1	0.9478	0.9543	0.9608	0.9596	0.9663
kappa	0.9229	0.9324	0.9419	0.9403	0.9501
model size	51.1 M	113 M	168 M	175 M	51.2 M

Figure 9 shows the test results of each network model on a partial verification set. As can be seen from the figure, the segmentation effect of the UNet and SegNet network models is poor. They have more misdivided pixels, and the effect of boundary segmentation is poor. The segmentation effect of UNet is slightly worse than that of SegNet. The network depth of the SUNet model is much smaller than that of the U2-Net and Deeplabv3+ networks. However, it can be seen from Figure 9 that its segmentation effect is better than that of the U2-Net and Deeplabv3+ networks, and the boundary segmentation effect is also better. The experimental results show that the segmentation effects of the five kinds of networks are as follows: UNet < SegNet < Deeplabv3+ < U^2-Net < SUNet.



(**a**) original image (**b**) mark (**c**) UNet (**d**) SegNet (**e**) U2-Net (**f**) DeepLabv3+ (**g**) SUNet

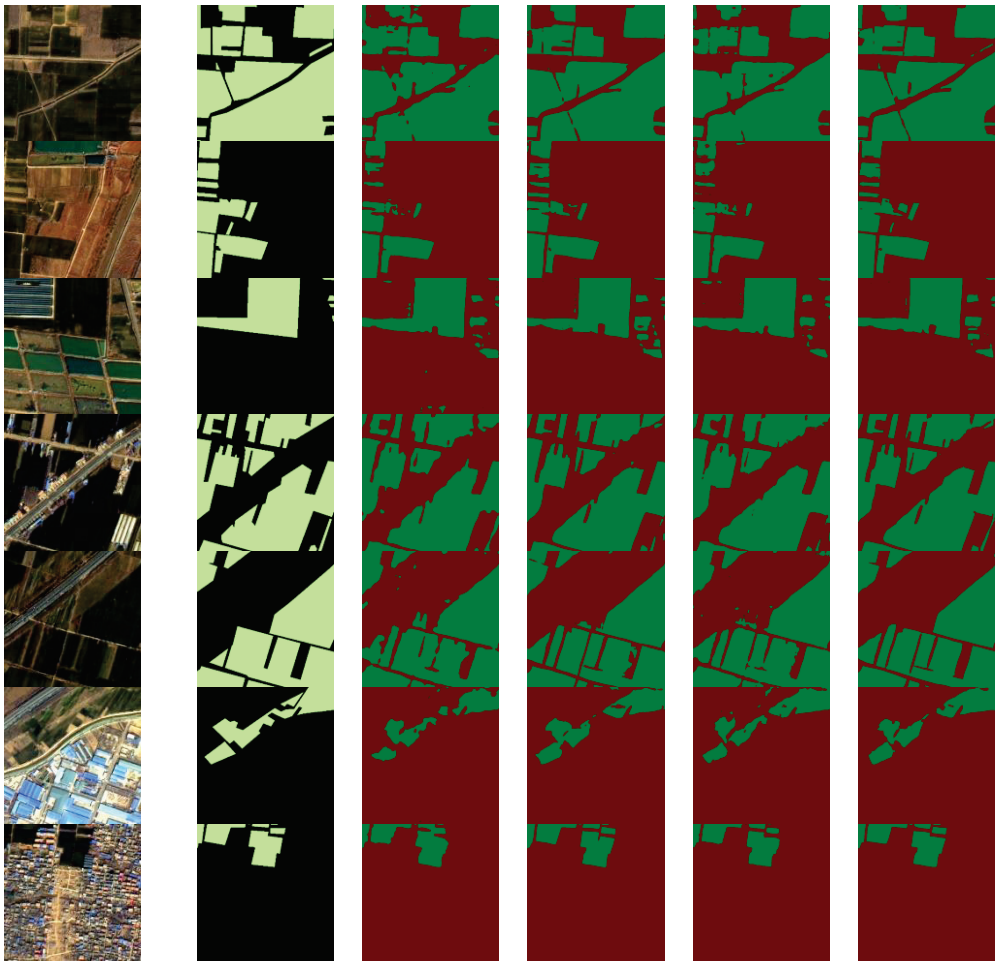
Figure 9. Experimental comparison results of winter wheat image dataset (Green areas are winter wheat, black areas and red areas are non-winter wheat).

3.2.2. Ablation Experiment Results

The extraction results after ablation experiments are shown in Figure 10, and the accuracy evaluation results are shown in Table 6. As can be seen from Table 6, whether it is Net-1 after adding the BN layer alone or Net-2 after adding the SA mechanism alone, all indicators have been improved compared with the original UNet network. However, the accuracy evaluation indexes of Net-1 and Net-2 are not as high as those of SUNet. Figure 10 shows the extraction results of the Net-1 and Net-2 network models on some verification sets. It can be seen from the graph that the segmentation effect of Net-1 and Net-2 are better than that of UNet, but both are slightly worse than that of SUNet.

Table 6. Evaluation index results of ablation experiment.

Index \ Model	UNet	Net-1	Net-2	SUNet
mIou	0.9261	0.9454	0.9361	0.9514
OA	0.9663	0.9753	0.9710	0.9781
precision	0.9460	0.9604	0.9516	0.9619
recall	0.9497	0.9634	0.9588	0.9707
F1	0.9478	0.9619	0.9552	0.9663
kappa	0.9229	0.9436	0.9337	0.9501



(**a**) original image (**b**) mark (**c**) UNet (**d**) Net-1 (**e**) Net-2 (**f**) SUNet

Figure 10. Extraction results of ablation experiment (Green areas are winter wheat, black areas and red areas are non-winter wheat).

4. Discussion

4.1. Comparative Test Discussion

The method used in this paper can identify and extract winter wheat in the western region of Lankao County, Kaifeng City. In addition, it can be seen from Table 5 and Figure 9 that SUNet's indicators have been improved. Compared with the UNet network, the SUNet network has a better recognition effect and higher recognition accuracy. Compared with the U^2-Net network proposed by Qin et al. [45] and the Deeplabv3+ network proposed by Chen et al. [46], the depth of SUNet is smaller, and the boundary segmentation effect of winter wheat is better. This shows that adding the BN layer and the SA mechanism can effectively alleviate the over-fitting phenomenon and shield some useless information to improve winter wheat recognition accuracy. From Figure 9, it can also be concluded that deep learning has a certain fault tolerance. Since the label is artificial, error and omission may inevitably occur, and the selection of samples is not completely correct. In the experiment, the label data were learned through deep learning, and the missing parts were sometimes identified. For example, a small part of winter wheat is not labeled on the third dataset's right side. However, all network models can be identified effectively. The fifth data group can be mistakenly classified as winter wheat, but the SUNet model performs well. This shows that the deep learning method has a strong fault tolerance, which can effectively reduce errors and improve target classification accuracy.

4.2. Ablation Experiment Discussion

From the results of Figure 10 and Table 6, the ablation experiment shows the following three points: Net-1 and Net-2 have a better extraction effect than UNet; Net-1′s extraction effect is better than that of Net-2. Finally, SUNet's extraction effect is better than that of Net-1 and Net-2. Firstly, by analyzing the comparative results between Net-1 and UNet, it can be observed that adding a BN layer can effectively prevent overfitting and improve segmentation accuracy. This is because including a BN layer allows the model to depend on statistical features during the training process, such as the mean and variance of each batch, rather than solely relying on input features. This approach can effectively control the model's complexity, thus avoiding overfitting. The comparison results between Net-2 and UNet show that the classification accuracy is improved after adding an SA mechanism. This is because the SA mechanism can enable the network to filter out irrelevant feature information, enhance the model's focus on regions of interest, and pay more attention to useful features, thereby improving classification accuracy. In addition, the comparison results of Net-1 and Net-2 indicate that adding a BN layer alone in UNet is more effective than adding an attention mechanism alone. This may be because the effect of the BN layer is more direct. In comparison, the advantage of the attention mechanism may not be as good as that of the BN layer when computing resources are limited. Finally, by analyzing the comparison results of SUNet, Net-1, and Net-2, it can be concluded that the performance of adding both a BN layer and attention mechanism in UNet is better than that of adding only a BN layer or attention mechanism separately. This is because a BN layer and attention mechanism can complement each other, improve the model's performance at different levels, and compensate for each other's shortcomings. This is also why SUNet has better feature extraction and accuracy evaluation indicators than other networks in this paper.

5. Conclusions

This work studied the recognition of winter wheat based on a convolutional neural network in the cloud environment. This paper improved the UNet model, proposed the SUNet model, and used the AI Studio cloud platform to study the application of convolution neural networks in winter wheat recognition on GF-2 high resolution. The conclusions are as follows: the high-performance environment equipped with an AI Studio cloud platform can provide efficient computing support for deep learning methods to identify winter wheat on high-resolution images. SUNet adds a BN layer and Shuffle Attention to UNet and uses the focus loss function to train the winter wheat dataset of multi-band GF-2 images with NDVI and $NDVI_{increase}$. This can effectively optimize the features. Thus, the precision of the winter wheat recognition on high-resolution images can be improved. The mIou calculated based on the confusion matrix is 0.9514. In addition, the overall classification accuracy, the F1 score of complete accuracy and the kappa coefficient are 0.9781, 0.9663 and 0.9501, respectively. Deep learning has a certain degree of fault tolerance, and unlabeled and mislabeled areas can be identified by training a convolution network. The work presented in this paper still has certain limitations. The dataset used in this study is suitable for extracting spatial information of winter wheat during a specific period. Currently, SUnet exhibits superior performance only in the task of spatial information extraction of winter wheat during specific periods. However, its performance in winter wheat change detection tasks, which involve detecting the variations of winter wheat at different time points, remains unclear. Therefore, further research and evaluation are needed to determine the performance and applicability of SUnet in the task of winter wheat change detection. The next step is to combine the time-series variations of the winter wheat growth cycle and extract the spatial distribution information of winter wheat during different periods.

Author Contributions: K.Z., R.M. and Y.Y. conceived the idea and proposed the method. L.L. provided suggestions for the research and helped edit the thesis. Z.Z., T.R. and M.Y. improved the quality of the manuscript and completed revisions. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by The Henan Science and Technology Project (202102210381), the Major Science and Technology Project of Kaifeng City (18ZD007), the Key Technology Project of Kaifeng City (2001001), and the Science and Technology Project of Henan Province (222102210061).

Data Availability Statement: The dataset used to support the findings of this study is provided at https://www.kaggle.com/datasets/zhangzzzy/winterwheatdataset accessed on 25 July 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zheng, W.Q. *Analysis of Wheat Price Formation Mechanism and Fluctuation Characteristics in China*; Chinese Academy of Agricultural Sciences: Beijing, China, 2016.
2. Chen, X.F.; Leeuw, G.; Arola, A.; Liu, S.; Liu, Y.; Li, Z.; Zhang, K. Joint retrieval of the aerosol fine mode fraction and optical depth using MODIS spectral reflectance over northern and eastern China: Artificial neural network method. *Remote Sens. Environ.* **2020**, *249*, 112006. [CrossRef]
3. Chen, X.F.; Zhao, L.M.; Zheng, F.J.; Li, J.; Li, L.; Ding, H.; Zhang, K.; Liu, S.; Li, D.; de Leeuw, G. Neural Network AEROsol Retrieval for Geostationary Satellite (NNAeroG) Based on T emporal, Spatial and Spectral Measurements. *Remote Sens.* **2022**, *14*, 980. [CrossRef]
4. Meng, S.Y.; Zhong, Y.F.; Luo, C.; Hu, X.; Wang, X.; Huang, S. Optimal Temporal Window Selection for Winter Wheat and Rapeseed Mapping with Sentinel-2 Images: A Case Study of Zhongxiang in China. *Remote Sens.* **2020**, *12*, 226. [CrossRef]
5. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; IEEE: New York, NY, USA; pp. 1–6.
6. Traore, B.B.; Kamsu-Foguem, B.; Tangara, F. Deep convolution neural network for image recognition. *Ecol. Inform.* **2018**, *48*, 257–268. [CrossRef]
7. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
8. Shi, X.R.; Zhou, F.; Yang, S.; Zhang, Z.J.; Su, T. Automatic Target Recognition for Synthetic Aperture Radar Images Based on Super-Resolution Generative Adversarial Network and Deep Convolutional Neural Network. *Remote Sens.* **2019**, *11*, 135. [CrossRef]
9. Gao, G.W.; Yu, Y.; Yang, J.; Qi, G.J.; Yang, M. Hierarchical Deep CNN Feature Set-Based Representation Learning for Robust Cross-Resolution Face Recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 2550–2560. [CrossRef]
10. Sladojevic, S.; Arsenovic, M.; Anderla, A.; Culibrk, D.; Stefanovic, D. Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Comput. Intell. Neurosci.* **2016**, *2016*, 3289801. [CrossRef]
11. Kang, K.; Ouyang, W.; Li, H.; Wang, X. Object detection from video tubelets with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 817–825.
12. Dhillon, A.; Verma, G.K. Convolutional neural network: A review of models, methodologies and applications to object detection. *Prog. Artif. Intell.* **2020**, *9*, 85–112. [CrossRef]
13. Liao, L.Y.; Du, L.; Guo, Y.C. Semi-Supervised SAR Target Detection Based on an Improved Faster R-CNN. *Remote Sens.* **2022**, *14*, 143. [CrossRef]
14. Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large kernel matters—Improve semantic segmentation by global convolutional network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4353–4361.
15. Mou, L.; Hua, Y.; Zhu, X.X. Relation matters: Relational context-aware fully convolutional network for semantic segmentation of high-resolution aerial images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 755–769. [CrossRef]
16. Yang, H.; Yu, B.; Luo, J.; Chen, F. Semantic segmentation of high spatial resolution images with deep neural networks. *GIScience Remote Sens.* **2019**, *58*, 749–768. [CrossRef]
17. Mo, Y.J.; Wu, Y.; Yang, X.N.; Liu, F.; Liao, Y. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing* **2022**, *493*, 626–646. [CrossRef]
18. Younis, M.C.; Keedwell, E. Semantic segmentation on small datasets of satellite images using convolutional neural networks. *J. Appl. Remote Sens.* **2020**, *13*, 046510. [CrossRef]
19. Yu, X.; Wu, X.; Luo, C.; Ren, P. Deep learning in remote sensing scene classification: A data augmentation enhanced convolutional neural network framework. *GIScience Remote Sens.* **2017**, *54*, 741–758. [CrossRef]
20. Fu, G.; Liu, C.J.; Zhou, R.; Sun, T.; Zhang, Q. Classification for High Resolution Remote Sensing Imagery Using a Fully Convolutional Network. *Remote Sens.* **2017**, *9*, 498. [CrossRef]

21. Zhang, W.; Tang, P.; Zhao, L.J. Remote Sensing Image Scene Classification Using CNN-CapsNet. *Remote Sens.* **2019**, *11*, 494. [CrossRef]
22. Tian, C.W.; Fei, L.K.; Zheng, W.X.; Xu, Y.; Zuo, W.; Lin, C.W. Deep Learning on Image Denoising: An Overview. *Neural Netw.* **2020**, *131*, 252–275. [CrossRef]
23. Izadi, S.; Sutton, D.; Hamarneh, G. Image denoising in the deep learning era. *Artif. Intell. Rev.* **2023**, *56*, 5929–5974. [CrossRef]
24. Liu, D.; Jia, J.L.; Zhao, Y.Q. Overview of Image Denoising Methods Based on Deep Learning. *Comput. Eng. Appl.* **2021**, *57*, 1–13.
25. Zhao, L.M.; Liu, S.M.; Chen, X.F.; Wu, Z.; Yang, R.; Shi, T.; Zhang, Y.; Zhou, K.; Li, J. Hyperspectral Identification of Ginseng Growth Years and Spectral Importance Analysis Based on Random Forest. *Appl. Sci.* **2022**, *12*, 5852. [CrossRef]
26. Chen, X.F.; Zheng, F.J.; Guo, D.; Wang, L.; Zhao, L.; Li, J.; Li, L.; Zhang, Y.; Zhang, K.; Xi, M.; et al. A review of machine learning methods for aerosol quantitative remote sensing. *J. Remote Sens.* **2021**, *25*, 2220–2233.
27. Zhang, C.; Gao, S.; Yang, X.; Li, F.; Yue, M.; Han, Y.; Zhao, H.; Zhang, Y.N.; Fan, K. Convolutional neural network-based remote sensing images segmentation method for extracting winter wheat spatial distribution. *Appl. Sci.* **2018**, *8*, 1981. [CrossRef]
28. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A deep convolutional encoderdecoderarchitecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
29. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv* **2017**, arXiv:1606.00915. [CrossRef] [PubMed]
30. Chen, Y. Research on Spatial Distribution Information Extraction of Winter Wheat Based on Multi-scale Feature Convolutional Neural Network. Master's Thesis, Shandong Agricultural University, Tai'an, China, 2021. [CrossRef]
31. Wang, S.Y.; Xu, Z.G.; Zhang, C.M.; Zhang, J.H.; Mu, Z.S.; Zhao, T.Y.; Wang, Y.Y.; Gao, S.; Yin, H.; Zhang, Z.Y. Improved winter wheat spatial distribution extraction using a convolutional neural network and partly connected conditional random field. *Remote Sens.* **2020**, *12*, 821. [CrossRef]
32. Lin, G.; Milan, A.; Shen, C.; Reid, I. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. *arXiv* **2016**, arXiv:1611.06612v3.
33. Teimouri, N.; Dyrmann, M.; Jørgensen, R.N. A novel spatio-temporal FCN-LSTM network for recognizing various crop types using multi-temporal radar images. *Remote Sens.* **2019**, *11*, 990. [CrossRef]
34. Zhou, K.; Liu, L.; Zhang, Y.N.; Miao, R.; Yang, Y. GEE-supported winter wheat area extraction and growth monitoring in Henan Province. *Sci. Agric. Sin.* **2021**, *54*, 2302–2318.
35. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.
36. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
37. Zhang, Q.L.; Yang, Y.B. Sa-net: Shuffle attention for deep convolutional neural networks. Proceeding of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; IEEE: New York, NY, USA; pp. 2235–2239.
38. Wu, Y.X.; He, K.M. Group Normalization. *Int. J. Comput. Vis.* **2020**, *128*, 742–755. [CrossRef]
39. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.
40. Chen, S.R.; Yan, L.; Shen, P.P. Design and Implementation of High-resolution Remote Sensing Satellite Image Preprocessing System. *Surv. Spat. Geogr. Inf.* **2021**, *44*, 1–3.
41. Liu, S.M.; Zhang, Y.L.; Zhao, L.M.; Chen, X.F.; Zhou, R.; Zheng, F.; Li, Z.; Li, J.; Yang, H.; Li, H.; et al. QUantitative and Automatic Atmospheric Correction (QUAAC) Application and Validation. *Sensors* **2022**, *22*, 3280. [CrossRef] [PubMed]
42. Tang, L.F.; Zhang, H.; Xu, H.; Ma, J.Y. A Survey of Image Fusion Methods Based on Deep Learning. *Chin. J. Image Graph.* **2023**, *28*, 3–36.
43. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2999–3007.
44. Wang, J.; Tian, H.; Wu, M.; Wang, L.; Wang, C. Rapid Remote Sensing Mapping of Winter Wheat in Henan Province. *J. Geo-Inf. Sci.* **2017**, *19*, 846–853.
45. Qin, X.; Zhang, Z.; Huang, C.; Dehghan, M.; Zaiane, O.R.; Jagersand, M. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognit.* **2020**, *106*, 107404. [CrossRef]
46. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

remote sensing

MDPI

Article

Multi-Objective Multi-Satellite Imaging Mission Planning Algorithm for Regional Mapping Based on Deep Reinforcement Learning

Yaxin Chen, Xin Shen *, Guo Zhang and Zezhong Lu

Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; chenyaxin@whu.edu.cn (Y.C.); guozhang@whu.edu.cn (G.Z.); luzezhong@whu.edu.cn (Z.L.)

* Correspondence: xinshen@whu.edu.cn

Abstract: Satellite imaging mission planning is used to optimize satellites to obtain target images efficiently. Many evolutionary algorithms (EAs) have been proposed for satellite mission planning. EAs typically require evolutionary parameters, such as the crossover and mutation rates. The performance of EAs is considerably affected by parameter setting. However, most parameter configuration methods of the current EAs are artificially set and lack the overall consideration of multiple parameters. Thus, parameter configuration becomes suboptimal and EAs cannot be effectively utilized. To obtain satisfactory optimization results, the EA comp ensates by extending the evolutionary generation or improving the evolutionary strategy, but it significantly increases the computational consumption. In this study, a multi-objective learning evolutionary algorithm (MOLEA) was proposed to solve the optimal configuration problem of multiple evolutionary parameters and used to solve effective imaging satellite task planning for region mapping. In the MOLEA, population state encoding provided comprehensive population information on the configuration of evolutionary parameters. The evolutionary parameters of each generation were configured autonomously through deep reinforcement learning (DRL), enabling each generation of parameters to gain the best evolutionary benefits for future evolution. Furthermore, the HV of the multi-objective evolutionary algorithm (MOEA) was used to guide reinforcement learning. The superiority of the proposed MOLEA was verified by comparing the optimization performance, stability, and running time of the MOLEA with existing multi-objective optimization algorithms by using four satellites to image two regions of Hubei and Congo (K). The experimental results showed that the optimization performance of the MOLEA was significantly improved, and better imaging satellite task planning solutions were obtained.

Keywords: regional mapping; satellite imaging mission planning; deep reinforcement learning; multi-objective optimization

1. Introduction

The regional image products obtained through satellite remote sensing are essential spatial data in the digital development of the region [1–3]. However, although remote sensing image processing has been widely researched for region mapping [4–6], limited studies have focused on satellite image acquisition technology. Therefore, the acquisition times of regional images have remained long, and periodic production remains difficult because precious satellite resources cannot be efficiently utilized. Imaging satellite mission planning is a crucial technology for improving the efficiency of satellite image acquisition [7]. Imaging benefits are maximized by appropriately adjusting satellite observation targets at suitable times according to user requirements, satellite attribute information, and relevant constraints, such as energy, mobility, and storage constraints. However, numerous contradictions exist between user requirements and imaging satellite performance. Therefore,

trade-offs exist in imaging satellite mission planning, which is a complex multi-objective optimization problem [8].

Numerous multi-objective evolutionary algorithms [9] have been proposed to achieve valuable results, such as many-objective optimization [10], large-scale multi-objective optimization [11], and parameter configuration [12]. With the development of evolutionary algorithms, parameter configuration has become critical in evolutionary computing because the performance of EAs depends considerably on their parameters [13]. However, it was often ignored because the sub-optimal parameter configuration can be compensated by prolonging the evolutionary generation and improving the evolutionary strategy. The parameter configuration methods are categorized into parameter tuning and parameter control. Parameter tuning refers to determining fixed parameter values for EAs before it runs. Manual adjustment is the most common method for parameter tuning [14,15]. Parameter tuning is mature and widely used in industry and academia. However, regardless of optimizing the parameter tuning method, a common shortcoming is as follows: the static parameter setting contradicts EAs, which is a dynamic process. Thus, the optimization performance of the EAs is limited.

Parameter control is a parameter configuration method in which the parameters change with the evolution process to improve algorithm performance. According to various parameter control methods, parameter control can be categorized into deterministic and adaptive parameter control.

Deterministic methods [16] are uninformed, i.e., population feedback information is not considered during the parameter selection. A time-varying function controls parameter selection in a deterministic method, such as the sigmoid [17] and piecewise functions [18]. Aiming at the parameter sensitivity problem of MOEA, Dai [19] proposed a deterministic MOEA framework based on collaborative granular sieving. The proposed method automatically selected evolutionary parameters guided by mathematical principles to guarantee convergence to the global optimal solution.

In adaptive parameter control [20,21], the magnitude and direction of evolution parameters are determined based on the partial feedback information. For example, in [22], instead of reference vectors, the optical solutions and normal-boundary directions were used to guide the population to adapt to the shape of the Pareto front automatically. In [23], an MOEA was proposed based on a self-adaptive crossover operator and interval initialization. Instead of the adaptive crossover rate, the proposed MOEA-ISa can dynamically determine the number of non-zero genes in offspring based on the similarity degree of parents. In [24], selection pressure was adapted by encoding the size of the tournament in the genome.

In recent years, some scholars used reinforcement learning (RL) [25] for parameter control. In this method, evolutionary parameters were selected through the feedback information of the population state, which is not affected by evolutionary strategies. Therefore, this method is a general parameter control method. In [26], the time difference algorithm was used to map the fitness value to evolutionary parameters (population size, competitive selection ratio, crossover rate, and mutation rate) by a Q table. However, because of the limitations of RL, only discrete small-scale optimization problems can be solved, and this method cannot be adapted to the continuous changes in evolutionary parameters. Therefore, the parameter control based on RL is not feasible. Deep reinforcement learning (DRL) [27] has been developed as a novel approach to overcome this problem. For example, Schuchardt uses DRL to solve the problem of single-parameter configuration for single objective functions [28]. Kaiwen transformed the multi-objective discrete problem into multiple single-objective discrete problems and then optimized these single-objective functions by DRL [29]. To solve the problem of sensitive parameter configuration of EAs, Song [30] proposed a genetic algorithm based on RL (RL-GA). In RL-GA, Q-learning was used for selecting crossover and mutation operations. Similarly, Tian [31] proposed a multi-objective optimization algorithm MOEA-DQN based on a deep Q network to solve the adaptive operator selection problem. In MOEA-DQN, four different mutation operators

were used as candidate operators for different population states. Reijnen [32] proposed a differential evolutionary (DE) algorithm based on DRL and parameter adaptive control, called DRL-APC-DE. The mutation parameters of the DE are adaptively controlled through DRL. To solve the scheduling problem of continuous annealing operations in the steel industry, Li [33] proposed an adaptive multi-objective differential evolutionary algorithm based on DRL, called AMODE-DRL. The AMODE-DRL can adaptively select mutation operators and parameters according to different search domains.

Although adaptive parameter configuration has been studied extensively, the following problems persist: first, the existing parameter control is based on a single parameter or a combination of multiple single parameters, which lacks the consideration of multiple parameters simultaneously. Furthermore, complex interactions exist among various parameters, and these influences are difficult to describe with mathematical expressions [13]. Second, evolutionary parameter configuration typically obeys the artificially set change rules or uses part of the population information. It results in the population information not accurately guiding the setting of evolutionary parameters, i.e., the parameter setting is sub-optimal. Third, the generality and robustness of the algorithm are unsatisfactory. The parameters of the same algorithm should be reconfigured for various application problems. Limited studies have focused on evolutionary parameter configuration methods based on DRL, configuring multiple evolutionary parameters simultaneously, adapting to continuous and discrete problems simultaneously, and general applicability to most MOEAs.

This study proposed a novel multi-objective learning evolutionary algorithm (MOLEA) based on DRL for satellite imaging task planning for regional mapping. The MOLEA maximizes the evolutionary benefits through DRL, thereby maximizing the optimization efficiency of the proposed algorithm. Specifically, the contribution of this study is summarized as follows:

(1) The scene data for fast training were constructed. For imaging satellite mission planning, the calculation is complex, and its efficiency is low, which results in a slow training speed for directly using the mission planning model as the training scene data. The easy-to-calculate public test functions were constructed as the same structure as the task planning model and then used as training scene data to improve the efficiency of algorithm training.
(2) Hypervolume (HV) was first proposed as a multi-objective evolutionary reward to guide the evolution of MOEA. As a scalar, HV makes it possible to apply DRL to MOEAs and enables the convergence and distribution of MOEAs to guide evolution simultaneously.
(3) The MOLEA was proposed. The MOLEA obtains all the population state information of each generation of the population through population state coding, which is used to guide the setting of evolutionary parameters. A simple neural network was used to map the relationship between population state information and multiple evolutionary parameters without determining its influence relationship. Through training, the parameter setting of each generation exhibits the maximum evolutionary benefit for improving the optimization performance of the algorithm.
(4) The effectiveness of the proposed MOLEA in imaging satellite mission planning was verified by comparing the MOLEA with four classical MOEAs. Experimental results revealed that the MOLEA considerably improves optimization efficiency and affects imaging satellite mission planning.

The rest of this paper is organized as follows. Section 2 briefly introduces the multi-satellite imaging task planning model for region mapping. Section 3 details the procedure of the proposed MOLEA. Section 4 presents the experimental results and analysis of the MOLEA and comparison algorithms. Finally, the discussion and conclusion are presented in Sections 5 and 6, respectively.

2. Multi-Satellite Mission Planning Model for Regional Mapping

We used the model in [34] for the multi-satellite imaging mission planning for area mapping as follows:

(1) **Decision variables:**

$$y_{ki} = \begin{cases} 1, \text{ if the strip } x_{ki} \text{participates in imaging} \\ 0, \text{ otherwise} \end{cases} \tag{1}$$

$$\mathbf{x} = (x_{11}, x_{12}, \cdots x_{ki} \cdots, x_{Kn}) \tag{2}$$

where y_{ki} indicates whether the strip with satellite swing angle x_{ki} participates in regional imaging. If so, $y_{ki} = 1$; otherwise, $y_{ki} = 0$. And $\mathbf{x}$ represents the set of swing angles for each imaging of the satellites, and its length is M. Furthermore, x_{ki} represents the imaging swing angle when satellite k passes the target area for the i-th time.

(2) **Objective functions:**

$$\min f(\mathbf{x}) = 1 - \frac{S_{\text{cov}}(\mathbf{x})}{S_{obj}} \tag{3}$$

$$\min g(y_{ki}) = \sum_{i=1}^{M} y_{ki} \Big/ M \tag{4}$$

where (3) ensures the maximum coverage of the imaging area. $S_{\text{cov}}(\mathbf{x})$ is the effective coverage area of all imaging strips, and S_{obj} is the target area. Equation (4) ensures the least number of satellite imaging, i.e., the least consumption of satellite resources.

(3) **Constraint conditions:**

$$06:00 \leq T_{\text{local}} \leq 18:00 \tag{5}$$

$$x_{\min} \leq x_{ki} \leq x_{\max} \tag{6}$$

where (5) indicates that the optical satellite satisfies the daytime imaging constraints, and (6) indicates the maneuverability of satellite k. $x_{\min}$ and $x_{\max}$ are the minimum and maximum swing angles of the satellite, respectively.

The multi-satellite imaging mission planning model used in this study is a real binary mixed coding multi-objective optimization problem.

3. Proposed MOLEA

3.1. MOLEA Framework

The MOLEA framework is displayed in Figure 1. The framework includes three core parts. First, scene data are constructed for fast training based on the public single-objective test function. Second, N-generation evolution is performed. This step is similar to the conventional evolution process. The difference is that evolutionary parameters are obtained by population state coding and neural network, and the evolutionary parameters need to be evaluated by calculating evolutionary reward, cumulative reward, and advantage estimation. Third, the evolutionary parameters are updated by proximal policy optimization (PPO) [35] based on the obtained training samples until the evolutionary parameter configuration with the greatest benefit is obtained.

3.2. Scene Data Construction for Fast Training

As we all know, the agent and the environment need to interact many times for DRL to generate sufficient training samples. Therefore, the mission planning model needs to be calculated numerous times. However, the mission planning model calculation involves region information, satellite orbit information, and complex imaging geometric operations. It is complex and inefficient. Therefore, there is a slow training speed by directly using the task planning model as the training scene data.

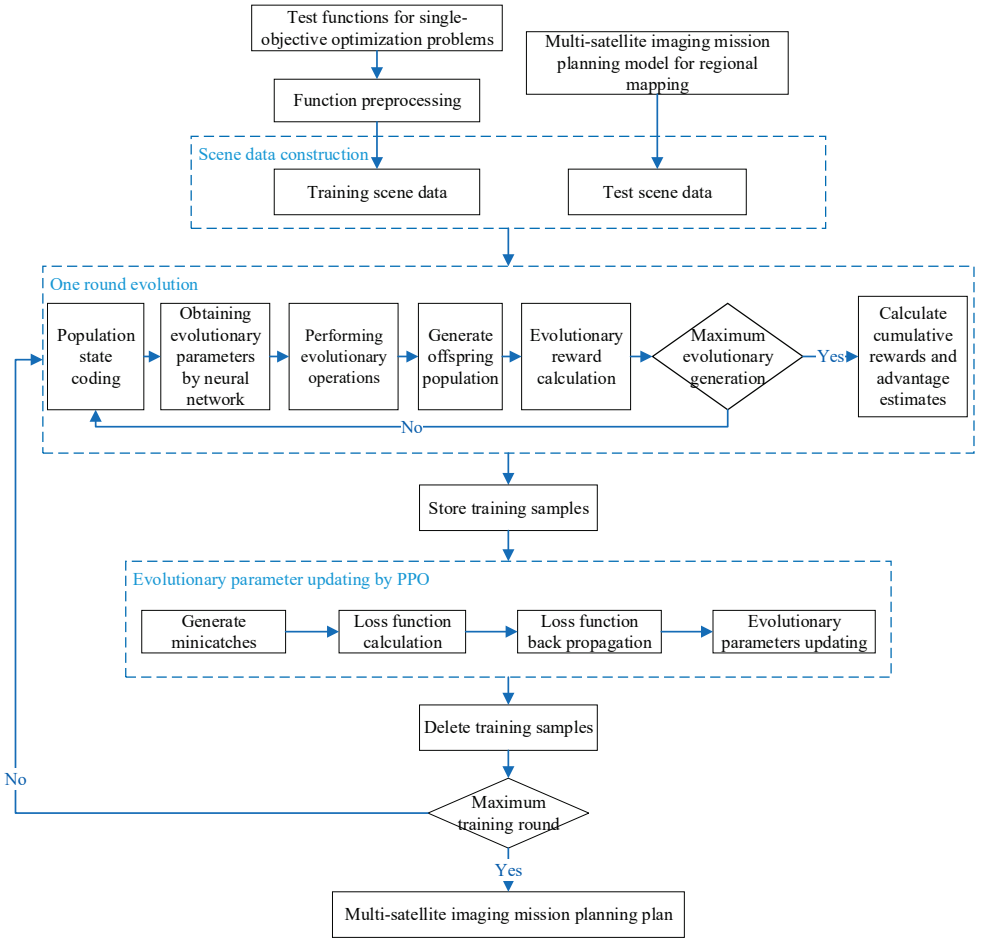


Figure 1. MOLEA framework.

A novel method is proposed to construct scene data based on the task planning model in Section 2 to improve the training speed. In this method, seven single-objective test functions with multi-dimensional decision variables are selected. Then, these functions are reconstructed through normalization and multi-objective transformation to have the same structure as the task planning model. The redeveloped functions are used as training scene data. The seven single-objective test functions are as follows: Griewank, rotated hyper-ellipsoid, quartic, Rastrigin, Rosenbrock, sphere, and Styblinski–Tang functions [36,37]. The multi-dimensional decision variables correspond to multiple imaging strips in the mission planning model. Normalization makes the value of $obj1$ changes within [0, 1], which corresponds to the first objective function of the task planning model. The multi-objective transformation adds a function $obj2$ to $obj1$, which corresponds to the second objective function of the task planning model. $obj2$ determines the decision variables involved in calculating $obj1$. The scene data constructed are as follows:

- Improved Griewank function:

$$\begin{cases} obj1 = \frac{1}{\max}\left(\sum\limits_{i=1}^{d}\frac{x_i^2}{4000} - \prod\limits_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1\right) \\ obj2 = \frac{\sum\limits_{i=1}^{d} y_i}{d} \end{cases} \tag{7}$$

- Improved rotated hyper-ellipsoid function:

$$\begin{cases} obj1 = \frac{1}{\max}\left(\sum_{i=1}^{d}\sum_{j}^{i} x_j^2\right) \\ obj2 = \frac{\sum_{i=1}^{d} y_i}{d} \end{cases} \tag{8}$$

- Improved quartic function:

$$\begin{cases} obj1 = \frac{1}{\max}\left(\sum_{i=1}^{n} i x_i^4 + random[0,1)\right) \\ obj2 = \frac{\sum_{i=1}^{d} y_i}{d} \end{cases} \tag{9}$$

- Improved Rastrigin function:

$$\begin{cases} obj1 = \frac{1}{\max}\left(10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)]\right) \\ obj2 = \frac{\sum_{i=1}^{d} y_i}{d} \end{cases} \tag{10}$$

- Improved Rosenbrock function:

$$\begin{cases} obj1 = \frac{1}{\max}\left(\sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]\right) \\ obj2 = \frac{\sum_{i=1}^{d} y_i}{d} \end{cases} \tag{11}$$

- Improved sphere function:

$$\begin{cases} obj1 = \frac{1}{\max}\left(\sum_{i=1}^{d} x_i^2\right) \\ obj2 = \frac{\sum_{i=1}^{d} y_i}{d} \end{cases} \tag{12}$$

- Improved Styblinski–Tang function:

$$\begin{cases} obj1 = \frac{1}{\max}\left(\frac{1}{2}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i)\right) \\ obj2 = \frac{\sum_{i=1}^{d} y_i}{d} \end{cases} \tag{13}$$

The constructed scene data exhibit the same structure as the task planning model and the same value range of decision variables and objective functions. Compared to the task planning model, the proposed constructed scene data calculation is simple and efficient. It effectively improves the training efficiency when the objective function requires many calculations.

3.3. Evolutionary Operations Based on DRL

The primary purpose of this section is to generate training samples for updating evolutionary parameters by performing evolutionary operations repeatedly. The training samples include the population state s_t of the t-th generation, evolutionary parameters of the t-th generation, new population state s_{t+1} reached, evolutionary reward R_t, cumulative reward $V(s_t)$, and advantage estimate $\hat{A}(t)$. In our research, the number of obtained training samples is the number of training scenes (7) $\times$ the number of optimizations for

each training scene (4) × the evolutionary generation for each optimization (100) × the total number of training iterations (500). The samples are obtained through the following steps.

3.3.1. Population State Coding

In order for the population information to fully influence the evolutionary parameters, the population state is encoded before the evolution. The selection of population states should consider all factors affecting evolutionary parameters and distinguish the differences between various populations. In this study, the following four features were used to code the population state:

- Feature 1: the real number solution of each individual in each generation of the population;
- Feature 2: the binary solution of each individual in each generation of the population;
- Feature 3: evolutionary reward for each generation of the population;
- Feature 4: the percentage of remaining generations $(T - t)/T$, where T is the total evolutionary generation, and t is the current generation.

Among them, Features 1 and 2 jointly describe each solution vector in the population. Feature 3 describes the reward value after the evolution is performed according to the current evolution parameters. Feature 4 describes the position of the current population in the evolution process and distinguishes various populations.

The data structure of the encoded population state information is displayed in Figure 2. In Figure 2, different colors represent different numerical values. To facilitate subsequent calculations, we extended Features 3 and 4 for consistency with the data dimensions of Features 1 and 2.

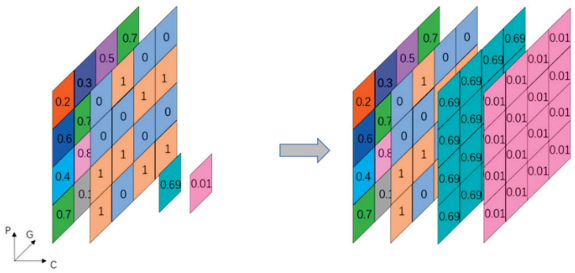


Figure 2. Population state coding.

3.3.2. Evolutionary Parameters Acquisition

The neural network in [28] was used to describe the relationship between the population state and multiple evolutionary parameters. However, in the neural network of this paper, multiple evolutionary parameters can be output simultaneously, whereas the neural network only output one evolutionary parameter in [28]. The neural network used in this study is displayed in Figure 3. The population state is the input, and the probability distribution parameters of evolution parameters are the output. The evolutionary parameters can then be obtained by sampling the probability distribution. The evolutionary parameters in this study include parent selection, real crossover, binary crossover, real mutation, and binary mutation parameters.

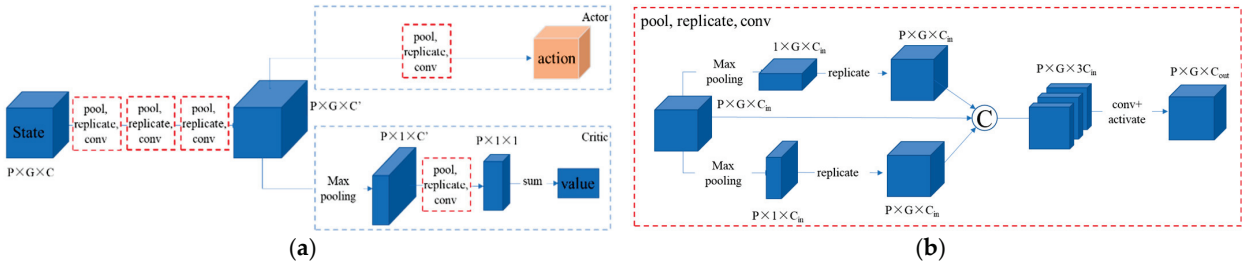


Figure 3. Neural network architecture. (**a**) The overall neural network architecture; (**b**) 'Pool, replicate, conv' sub-structure of the neural network.

For the parent selection parameter, the Bernoulli distribution is used:

$$X \sim B(n, p) \tag{14}$$

where p is the output parameter of the neural network, $n = 1$, and the sampling value of the Bernoulli distribution is 0 or 1 for determining whether the individual in the population is selected as the parent.

For other parameters, beta distribution is adopted:

$$X \sim Be(\alpha, \beta) \tag{15}$$

For each evolutionary parameter, α and β are the outputs of the neural network, and the value range of beta distribution is [0, 1].

The output of the neural network includes nine elements:

$$actor = \{p, \alpha_{real_c}, \beta_{real_c}, \alpha_{bin_c}, \beta_{bin_c}, \alpha_{real_m}, \beta_{real_m}, \alpha_{bin_m}, \beta_{bin_m}\} \tag{16}$$

As shown in Figure 3, another output of the neural network, critic, is used to evaluate how good an actor is in DRL. Critic and actor operate in the same environment and exhibit similar features. Therefore, the actor and critic share low-level features and then output actor and critic, respectively.

3.3.3. Evolutionary Reward, Cumulative Reward, and Advantage Estimation

The evolutionary reward can be calculated after performing one-generation evolution according to the obtained evolutionary parameters. For a single-objective function, the difference between the maximum fitness values in the two generations of populations can be used as the evolutionary reward. However, it is difficult to calculate the fitness and the evolutionary reward for the two-dimensional multi-objective function because multiple Pareto solutions do not dominate each other. This paper innovatively uses the difference between the HV of the two generations of populations as the evolutionary reward.

HV represents the volume of the space surrounded by Pareto individuals and the reference point. The schematic of the HV calculation is shown in Figure 4. Because the HV calculation does not require a true Pareto solution, it is suitable for evaluating practical application problems, e.g., imaging satellite mission planning.

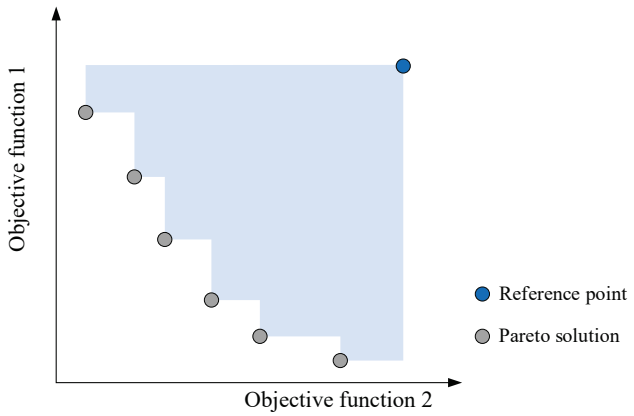


Figure 4. Schematic of HV calculation.

The calculation formula of evolutionary reward based on HV is as follows:

$$R_t = HV_t - HV_{t-1} \tag{17}$$

where HV_t and HV_{t-1} represent the HV value of generations t and $t-1$, respectively, and R_t represents the evolutionary reward of the t-th generation.

The larger the HV, the better the convergence and distribution of the solution, i.e., the better the quality of the solution.

The evolutionary reward is used to evaluate the evolutionary gain of one-generation evolution. In this study, we should not only make the evolutionary parameters output by the neural network produce the maximum benefit in contemporary evolution but also in future generations. The cumulative reward is the sum of the benefits generated by the current evolutionary parameters in future generations. This value indicates that the evolutionary parameters have foresight.

$$V(s_t) = \sum_{i=0}^{T-t+1} \gamma^i R_{t+i} \tag{18}$$

where γ is the discount factor. It represents the loss of the current evolutionary parameters to generate a reward in the future, and $\gamma \in [0, 1]$.

To evaluate the rationality of selecting evolutionary parameter *actor* in the state s_t, we used the advantage estimate [38]. It is calculated as follows:

$$\hat{A}_t = \hat{Q}(s_t, a_t) - \hat{V}(s_t) = \sum_{i=0}^{T-t+1} (\gamma\lambda)^i \delta_{t+i} \tag{19}$$

$$\delta_t = \gamma \hat{V}(s_{t+1}) + R_t - \hat{V}(s_t) \tag{20}$$

where $\hat{Q}(s_t, a_t)$ represents the reward of taking the actor a_t under the state s_t. $\hat{V}(s_t)$ represents the average reward of all actions under state s_t. $\hat{V}(s_t)$ is the critic obtained by the neural network. The parameter $\lambda \in [0, 1]$ controls the trade-off between variance and bias of the advantage estimate; δ_t represents the difference between the calculated value by (18) and the estimated value of the cumulative reward of the t-th generation by the neural network.

Furthermore, $\hat{A}_t > 0$ denotes that the current action is superior to the average action. The higher the probability of action is, the better; otherwise, $\hat{A}_t < 0$, the smaller the probability of action is, the better. By increasing the probability of good actions, the evolutionary algorithm can improve evolutionary gains.

3.4. Evolutionary Parameters Updates Using PPO

In the MOLEA, the PPO is used to update the evolutionary parameters for two reasons: first, the PPO introduces importance sampling, so that the sampled data obtained by the interaction can be used for policy updates many times. Thus, data sampling efficiency is improved. Second, in PPO, first-order optimization is used to constrain the changed extent of the difference between the policy interacting with the environment and the policy to be updated. It makes the difference change within a specific range to ensure the stability of the DRL. PPO achieved an excellent balance between data sampling efficiency, algorithm performance, and debugging complexity.

The loss function of the PPO is expressed as follows:

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\Big[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\Big] \tag{21}$$

In (21), $L_t^{CLIP}(\theta)$ is the first-order optimization clipping loss function and calculated as follows:

$$L_t^{CLIP}(\theta) = \hat{\mathbb{E}}_t\big[\min\big(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t\big)\big] \tag{22}$$

In (22), $\hat{\mathbb{E}}_t[\cdots]$ represents the expected value of the sample. $r_t(\theta)$ represents the ratio between the strategy interacting with the environment $\pi_{\theta_{old}}(a_t|s_t)$ and the strategy to be updated $\pi_\theta(a_t|s_t)$. The numerical meanings of $\pi_\theta(a_t|s_t)$ and $\pi_{\theta_{old}}(a_t|s_t)$ are the probability

of taking action a_t in the state s_t under the probability distribution described by strategies π_θ and $\pi_{\theta_{old}}$, respectively.

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{23}$$

The clipping function $clip(\cdot)$ renders $r_t(\theta)$ change within $(1-\varepsilon, 1+\varepsilon)$ to ensure that the policies $\pi_\theta(a_t|s_t)$ and $\pi_{\theta_{old}}(a_t|s_t)$ do not differ considerably.

In (21), $L_t^{VF}(\theta)$ represents the mean squared error (MSE) between the neural network estimate $\hat{V}(s_t)$ and the value function $V(s_t)$:

$$L_t^{VF}(\theta) = \mathbb{E}\left[\left\|\hat{V}_\theta(s_t) - V(s_t)\right\|^2\right] \tag{24}$$

In (21), $S[\pi_\theta](s_t)$ represents the exploration based on maximum information entropy, which is used to prevent the algorithm from falling into the local optimum.

After multiple rounds of repeated training, the evolutionary parameters are continuously updated in the direction with more excellent benefits until the parameters' configuration policy for the maximum benefit is obtained, i.e., the algorithm optimization efficiency reaches the highest point.

4. Experimental Results and Analysis

4.1. Experimental Settings

(1) Training Function

The training function is the set of seven scene data introduced in Section 3.2.

(2) Test Function

The test functions are the imaging satellite mission planning models introduced in Section 2 for Hubei and Congo (K). Imaging satellites include four optical satellites, namely GF1, GF6, ZY1-02C, and ZY3, carrying the 2-m load. The parameters of each satellite are presented in Table 1. The imaging completion time was from September 8 to 20, 2019, i.e., 13 days. Within the imaging completion time, four satellites passed through Hubei and Congo (K) 14 and 40 times, respectively. Therefore, the number of decision variables of imaging satellite mission planning models in Hubei and Congo (K) were 14 and 40, respectively.

Table 1. Experimental Satellites and Corresponding Sensor Parameters.

Satellite	GF1	GF6	ZY1-02C	ZY3
Launch time	26 Apr. 2013	2 Jun. 2018	22 Dec. 2011	9 Jan. 2012
Orbit type	Sun-synchronous return orbit			
Orbital altitude (km)	645	645	780	506
Regression cycle (day)	41	41	55	59
Swing ability (°)	±35	±35	±25	±32
Imaging width (km)	60	90	54	51
Half field of view (°)	2.67	3.99	1.98	2.88

(3) Parameter setting

Two types of parameters were used: one is the set of EA-related parameters, and the other is the set of DRL super parameters.

For the EA-related parameters, there are evolutionary generation and population size besides the parameters that need to be learned in MOLEA. The population size and evolutionary generation were set to 12 and 100 during the training. During the test, the evolutionary generation and population size settings differed according to experimental requirements. The EA-related parameters are presented in Table 2, and the super parameters of the PPO are detailed in Table 3.

Table 2. Population Size and Evolutionary Generation Settings.

Process	Population Size	Evolutionary Generation	Number of Decision Variables
Training	12	100	14
Test	12/20/40	100/200	14/40

Table 3. Hyperparameter Settings for PPO.

Hyperparameter	Value	Hyperparameter	Value
Learning rate	10^{-4}	λ	0.99
Batch size	400	Clipping parameters ε	0.2
Epochs	4	Ratio factor c_1	0.5
Optimizer	Adam	Ratio factor c_2	10^{-4}
Discount factor	0.99	-	-

4.2. Comparison of Results of Various MOEAs

In this experiment, the training results obtained from seven scene data were applied to mission planning for Hubei and compared with four classic algorithms, NSGA-II [39], MOEA/D [40], MOPSO [41], and SPEA2 [42]. The population size, evolutionary generation, and the number of decision variables in MOLEA testing were 12, 100, and 14, respectively. These parameters of comparison algorithms were the same. The distributions of non-dominated solutions obtained by various algorithms are displayed in Figure 5, and the specific solution values are listed in Table 4.

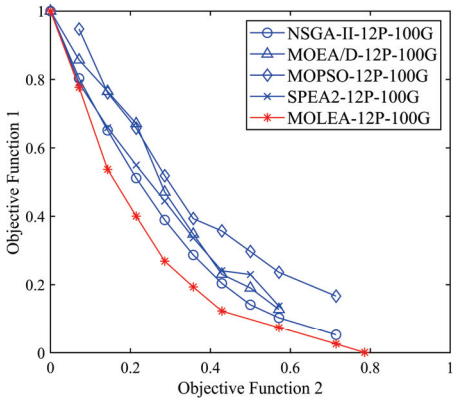


Figure 5. Distribution of non-dominated solutions of various algorithms for mission planning in Hubei.

Table 4. Values of Non-dominated Solutions of Various Algorithms of Mission Planning in Hubei.

Non-Dominated Solution	NSGA-II		MOEA/D		MOPSO		SPEA2		MOLEA	
	obj 1	obj 2	obj 1	obj 2	obj 1	obj 2	obj 1	obj 2	obj 1	obj 2
1	0.28699	0.35714	1.00000	0.00000	0.94767	0.07143	0.33694	0.35714	1.00000	0.00000
2	1.00000	0.00000	0.85721	0.07143	0.39328	0.35714	0.13741	0.57143	0.00132	0.78571
3	0.38934	0.28571	0.76516	0.14286	0.35726	0.42857	0.24065	0.42857	1.00000	0.00000
4	0.20402	0.42857	0.67111	0.21429	0.76134	0.14286	0.54970	0.21429	0.53714	0.14286
5	0.10277	0.57143	0.47102	0.28571	0.29695	0.50000	0.79007	0.07143	0.07386	0.57143
6	0.14116	0.50000	0.34808	0.35714	0.23636	0.57143	0.65933	0.14286	0.40033	0.21429
7	0.80333	0.07143	0.23057	0.42857	0.16650	0.71429	1.00000	0.00000	0.12276	0.42857
8	0.05223	0.71429	0.19044	0.5	0.65756	0.21429	1.00000	0.00000	0.26877	0.28571
9	0.51186	0.21429	0.12749	0.57143	0.51871	0.28571	0.22989	0.50000	0.02606	0.71429
10	0.05223	0.71429	-	-	-	-	0.79007	0.07143	0.77720	0.07143
11	1.00000	0.00000	-	-	-	-	0.44456	0.28571	0.19321	0.35714
12	0.65126	0.14286	-	-	-	-	0.65933	0.14286	0.77720	0.07143

Note: The gray shadows mark the solution with maximum coverage for each MOEA.

In Figure 5, the abscissa is the value of objective function 2, which denotes the percentage of the participating imaging strips in all strips. The smaller the value is, the lesser the number of strips required to cover the area, i.e., the higher the utilization rate of satellite resources. The ordinate is the value of objective function 1, which denotes the difference between 1 and the imaging coverage rate. The smaller the value is, the higher the coverage rate of the target area is. The gray shading in Table 4 indicates the solution with the largest coverage.

Figure 5 reveals that the distribution and convergence of the planning solutions obtained by the MOLEA are superior to the comparison algorithms. Under the same population size and evolutionary generation, the MOLEA obtains higher coverage when the number of participating imaging strips is the same, and the required imaging strips are less when the coverage is the same. Table 4 reveals that the highest coverage rate of the MOLEA is 99.87%, whereas the coverage rates of the four comparison algorithms are 94.78%, 87.25%, 83.35%, and 86.25%, respectively. This is because the proposed parameter configuration method effectively selects better evolutionary parameters than existing methods based on different population states. It significantly improves the performance of MOLEA. The results reveal that the performance of the MOLEA is superior to the four comparison algorithms under the same population size and evolutionary generation.

4.3. Comparison of Multiple Training Results

To verify the MOLEA reliability, the same parameters were used to train ten agents. The training results were used for planning Hubei. The planning results were compared with NSGA-II, which has the best-performing among the comparison algorithms. The result is displayed in Figure 6, where red is the distribution of non-dominated solutions obtained by NSGA-II, and black is the distribution of non-dominated solutions obtained by 10 agents of the MOLEA. Table 5 lists the solutions with the highest coverage of the 10 agents of the MOLEA and the four comparison algorithms.

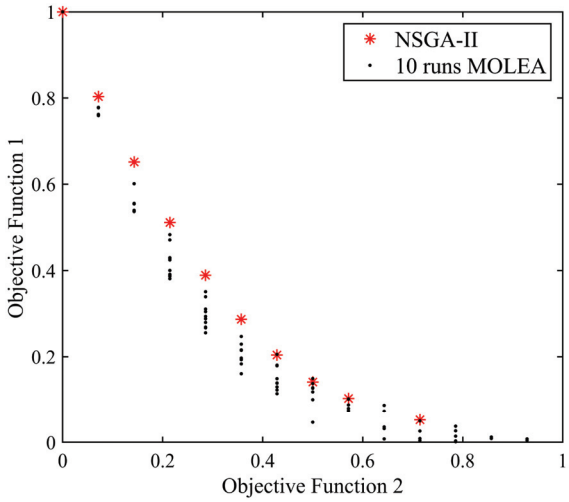


Figure 6. Distribution of non-dominated solutions for different agents in Hubei.

Table 5. Solutions with Maximum Coverage for Different Agents and algorithms.

Agent Number	Obj 1	Coverage Rate	Obj 2	Number of Imaging Strips
1	0.00876	99.12%	0.85714	12
2	0.00777	99.22%	0.64286	9
3	0.00402	99.60%	0.71428	10
4	0.00315	99.68%	0.78571	11
5	0.00428	99.57%	0.92857	13
6	0.00132	99.87%	0.78571	11

Table 5. *Cont.*

Agent Number	Obj 1	Coverage Rate	Obj 2	Number of Imaging Strips
7	0.01209	98.79%	0.85714	12
8	0.01412	98.59%	0.78571	11
9	0.00776	99.22%	0.92857	13
10	0.00863	99.14%	0.71428	10
NSGA-II	0.05223	94.78%	0.71429	10
MOEA/D	0.12749	87.25%	0.57143	8
MOPSO	0.16650	83.35%	0.71428	10
SPEA2	0.13741	86.26%	0.57143	8

In Figure 6, the solution of the MOLEA is generally under the solution of NSGA-II. This means that the distribution and convergence of MOLEA are better than NSGA-II. The MOLEA is highly reliable. Table 5 reveals that the coverage of 10 agents is higher than that of the comparison algorithms. Among the ten agents, eight agents have the coverage rate of more than 99%, and the other two agents have the coverage rate of more than 98%. Among the comparison algorithms, the algorithm with the highest coverage rate is NSGA-II, whose coverage rate is 94.78%, whereas the coverage rate of the worst-performing MOPSO is only 83.35%. For ten different agents, better results can always be obtained than by comparing algorithms. This is because the evolutionary parameter selection of the MOLEA always aims to maximize future evolutionary benefits. Therefore, ten agents can always perform well in different population states. It is further verified that the proposed MOLEA has higher reliability and stronger ability to obtain the optimal solution than comparison algorithms.

4.4. Comparison of Evaluation Index HV

The reference point is selected (1.1, 1.1) because the maximum values of objective functions 1 and 2 in the mission planning model are 1. Figure 7 displays the changes in the HV with evolutionary generations. The solid red line is the changed HV of the MOLEA, and the dotted gray, orange, green, and blue lines represent the changed HV of the comparison algorithms.

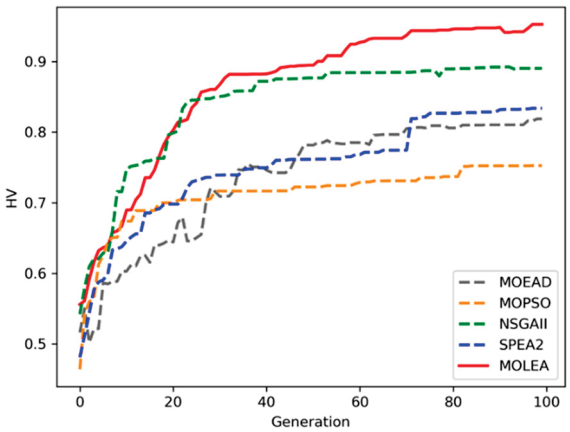


Figure 7. Changes in HV of various algorithms with evolutionary generations.

Figure 7 reveals that the HV of each algorithm increases gradually in the first twenty generations. NSGA-II and MOLEA exhibit approximate performance, with the largest increase and the best performance, followed by MOPSO and SPEA2, which are superior to MOEA/D. Each algorithm exhibits distinct characteristics with the evolution process: although the HV of MOEA/D fluctuates, the value continues to increase and gradually surpasses MOPSO and is close to the HV of SPEA2. Although MOPSO exhibits good growth in the early stage, the change in HV gradually became stable after the twentieth generation. The increase was not obvious, i.e., MOPSO fell into a local optimum. For SPEA2, the HV

growth is slow until a step-like growth occurs close to the seventy-sixth generation and then tends to change steadily. NSGA-II performs best among all comparison algorithms, consistent with the above experimental results. However, NSGA-II exhibits apparent growth in the first 40 generations of evolution, and the change tends to be stable after 40 generations. To some extent, this means that the NSGA-II falls into the local optimum, and the searching efficiency for the optimal solution becomes slow. However, the HV of MOLEA continues to increase during the evolution process, and it continues to increase in a slight range even in the subsequent evolution, i.e., it continues to search for the optimal solution effectively. These experimental results revealed that the optimization efficiency of the learning evolution strategy of the MOLEA is considerably higher than that of the random search strategy of the comparison algorithms. After the evolution was completed, the HV of the MOLEA was considerably higher than the comparison algorithms. This result indicated that the convergence and distribution of the optimal solution of the MOLEA were better than comparison algorithms.

4.5. Comparison of Results of Various Evolutionary Generations and Different Population Sizes

To further verify the effectiveness of the MOLEA, this experiment increases the evolutionary generation or the population size of the comparison algorithms. The new results were compared with the result of the MOLEA in Section 4.2. Figures 8 and 9 show the distribution of planning solutions in Hubei when the comparative algorithms increase the evolutionary generation and population size, respectively. Tables 6 and 7 show the solutions with the maximum coverage rate of each algorithm under various evolutionary generations and population sizes, respectively.

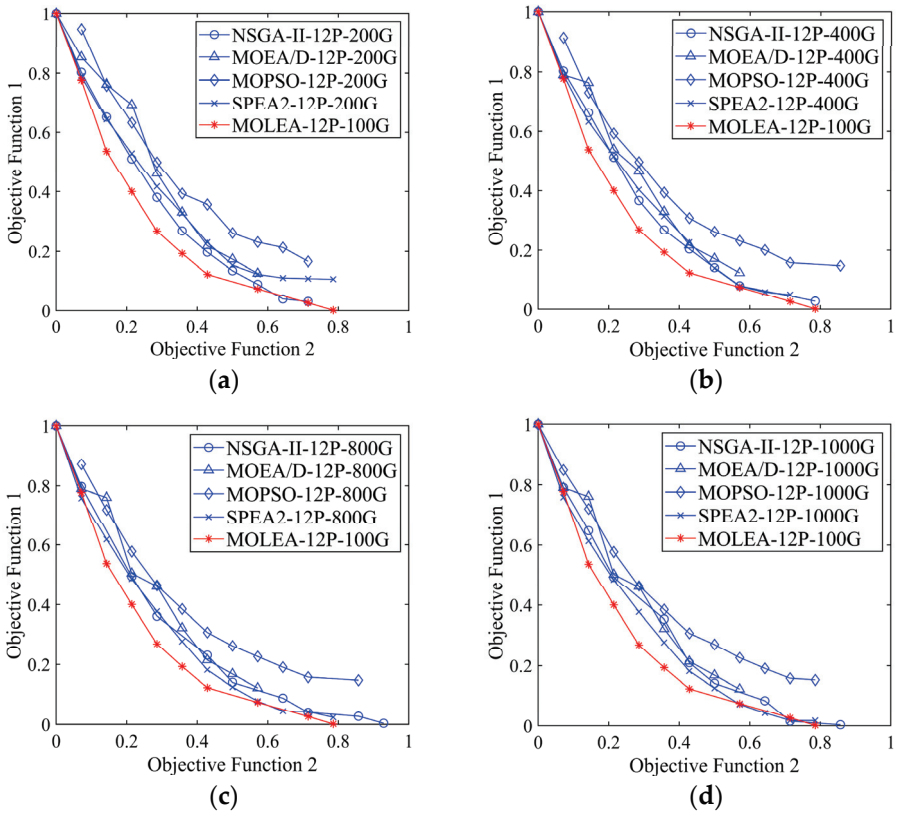


Figure 8. Distribution of non-dominated solutions of various algorithms with the same population size 12 but different evolutionary generations. MOLEA evolution for 100 generations. (**a**) Comparison algorithms evolution for 200 generations. (**b**) Comparison algorithms evolution for 400 generations. (**c**) Comparison algorithms evolution for 800 generations. (**d**) Comparison algorithms evolution for 1000 generations.

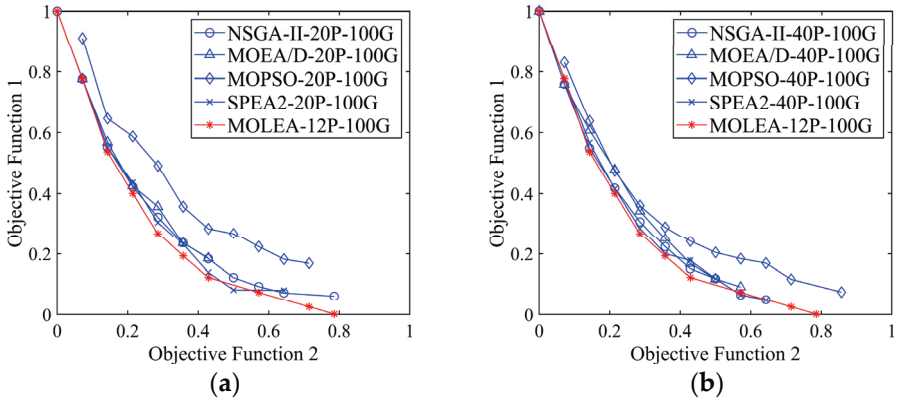


Figure 9. Distribution of non-dominated solutions for various algorithms in the same evolutionary generation 100 but with different population sizes. The population size of the MOLEA is 12. (**a**) The population size of the comparison algorithms is 20. (**b**) The population size of the comparison algorithms is 40.

Table 6. Solutions with the Maximum Coverage for Different Algorithms in Different Evolutionary Generations.

	Obj 1	Coverage Rate	Obj 2	Number of Imaging Strips
NSGA-II-12P-200G	0.03106	96.89%	0.71429	10
NSGA-II-12P-400G	0.02678	97.32%	0.78571	11
NSGA-II-12P-800G	0.00365	99.63%	0.92857	13
NSGA-II-12P-1000G	0.00316	99.68%	0.85714	12
MOEA/D-12P-200G	0.12630	87.37%	0.57143	8
MOEA/D-12P-400G	0.12283	87.72%	0.57143	8
MOEA/D-12P-800G	0.12192	87.81%	0.57143	8
MOEA/D-12P-1000G	0.12187	87.81%	0.57143	8
MOPSO-12P-200G	0.16650	83.35%	0.71429	10
MOPSO-12P-400G	0.14728	85.27%	0.85714	12
MOPSO-12P-800G	0.14728	85.27%	0.85714	12
MOPSO-12P-1000G	0.15232	84.77%	0.78571	11
SPEA2-12P-200G	0.10671	89.33%	0.78571	11
SPEA2-12P-400G	0.04927	95.07%	0.71429	10
SPEA2-12P-800G	0.02357	97.64%	0.78571	11
SPEA2-12P-1000G	0.01727	98.27%	0.78571	11
MOLEA-12P-100G	0.00132	99.87%	0.78571	11

Note: The gray shadows mark the solutions of comparative algorithms with coverage exceeding 98%.

Table 7. Solution with the Maximum Coverage under Various Population Sizes.

	Obj 1	Coverage Rate	Obj 2	Number of Imaging Strips
NSGA-II-20P-100G	0.06149	93.85%	0.78571	11
MOEA/D-20P-100G	0.18579	81.42%	0.42857	6
MOPSO-20P-100G	0.16898	83.10%	0.71428	10
SPEA2-20P-100G	0.08059	91.94%	0.64286	9
NSGA-II-40P-100G	0.04968	95.03%	0.64286	9
MOEA/D-40P-100G	0.09161	90.84%	0.57143	8
MOPSO-40P-100G	0.07511	92.49%	0.85714	12
SPEA2-40P-100G	0.04892	95.11%	0.64286	9
MOLEA-12P-100G	0.00132	99.87%	0.78571	11

Note: The gray shadows mark the two solutions of comparative algorithms with maximum coverage.

The following conclusions can be drawn from Figure 8: First, from the distribution of solutions, the results of the MOLEA for 100 generations are still better than the results of the comparison algorithms for 1000 generations. Second, each comparison algorithm converges to a better solution as the evolutionary generation increases, but the computational cost exceeds the benefit. Third, NSGA-II exhibited the best performance among the comparison algorithms only from the solution with the maximum coverage, followed by SPEA2. As presented in Table 6, more than 98% coverage can be achieved when NSGA-II evolves for 800 and 1000 generations, or SPEA2 evolves for 1000 generations. They are close to the MOLEA evolutions for 100 generations.

From Table 7 and Figure 9, the following conclusions can be drawn: First, as the population size increases, the solution distributions of the comparison algorithms improve gradually. Second, although the comparison algorithms gradually improved in the distribution of solutions, they were still inferior to the MOLEA in obtaining the optimal solution with the maximum coverage, even though the comparison algorithms have a larger population size. In this experiment, the two comparative algorithms with the highest coverage rate were NSGA-II and SPEA2 when the population size was 40, and their corresponding coverage rates were 95.03% and 95.11%, respectively. However, the coverage rate of the MOLEA was more than 98% when the population size was 12. By contrast, the computational consumption of NSGA-II and SPEA2 increased by about 70%.

By increasing the evolutionary generation or population size, comparative algorithms achieved better results. This is because the optimization performance of the comparison algorithms is so inefficient that they did not achieve the optimal performance in Section 4.2. The performance of comparative algorithms that increase evolutionary generation or population size is still inferior to that of MOLEA. This is because the proposed autonomous parameter configuration method that maximizes future returns greatly stimulates algorithm performance, enabling MOLEA to quickly find the optimal solution. At the same time, it also verifies the fact that evolutionary parameters have an important impact on evolutionary algorithms.

4.6. Algorithm Running Time Analysis

Table 8 details the running time of each algorithm for various population sizes and evolutionary generations. The two conclusions can be drawn from Table 8: First, from the same population size and evolutionary generation (12 and 100), the running time of the MOLEA is 11.25 s, which is only better than the 51.28 s of the MOEA/D. The main reason is that the evolution parameters of the comparison algorithms were fixed, while the evolution parameters of the MOLEA were obtained by sampling from various distributions according to the trained network. Therefore, the MOLEA required more time consumption. Second, from obtaining evolutionary results, 11.25 s were for the MOLEA to reach more than 98% coverage, whereas the time for NSGA-II running 800 generations (99.63% coverage) was 66.23 s, and for NSGA-II running 1000 generations (99.68% coverage) it was 81.12 s, and for SPEA2 running 1000 generations (98.27% coverage) it was 82.0 s. The MOLEA exhibits apparent advantages in running time. This is because the excellent performance of the MOLEA enables it to obtain the optimal solution in a short time. However, comparing algorithms to obtain solutions of the same quality requires more time due to their limited optimization ability.

Table 8. Running Time of Each Algorithm at Different Population Sizes and Evolutionary Generations (s).

Population Size	Evolutionary Generation	NSGA-II	MOEA/D	MOPSO	SPEA2	MOLEA
12	100	8.64	51.28	8.59	8.21	11.25
12	200	16.38	102.36	16.79	16.25	-
12	400	32.73	206.28	33.89	32.94	-
12	800	66.23	411.5	68.34	63.81	-
12	1000	81.12	512.22	84.44	82.0	-
20	100	11.06	87.09	11.79	11.00	-
40	100	17.06	172.04	19.32	17.91	-

Note: The gray shadows mark the running time required for each MOEA to obtain solutions with coverage exceeding 98%.

4.7. Planning Results of the Same Training Data in Congo (K)

In this experiment, the training results in Section 4.2 were used in planning Congo (K), which has more imaging strips. Figure 10 displays the distribution of non-dominated solutions in the Congo (K) under various evolutionary generations and population sizes. Table 9 lists the solutions with the maximum coverage rate of each algorithm under various population sizes and evolutionary generations.

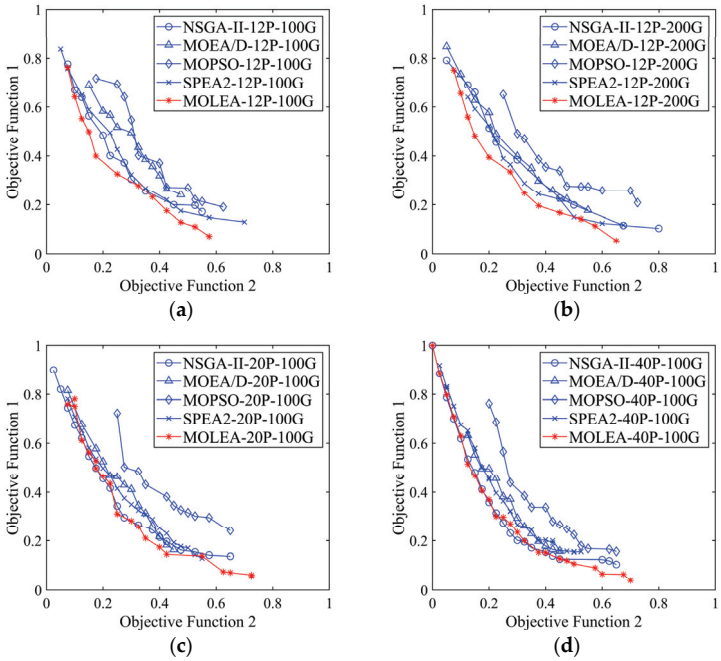


Figure 10. Distribution of non-dominated solutions at various population sizes and evolutionary generations in Congo (K). (**a**) population size 12 and 100 generations. (**b**) population size 12 and 200 generations. (**c**) population size 20 and 100 generations. (**d**) population size 40 and 100 generations.

Table 9. Mission Planning Solutions with The Maximum Coverage in Congo (K) by Various Algorithms under Different Population Sizes and Evolutionary Generations.

	Obj 1	Coverage Rate	Obj 2	Number of Imaging Strips
NSGA-II-12P-100G	0.17188	82.81%	0.55	22
MOEA/D-12P-100G	0.24126	75.87%	0.475	19
MOPSO-12P-100G	0.19061	80.94%	0.625	25
SPEA2-12P-100G	0.13094	86.91%	0.7	28
MOLEA-12P-100G	0.07328	92.67%	0.575	23
NSGA-II-12P-200G	0.10514	89.49%	0.8	32
MOEA/D-12P-200G	0.17795	82.21%	0.55	22
MOPSO-12P-200G	0.20814	79.19%	0.725	29
SPEA2-12P-200G	0.11842	88.16%	0.675	27
MOLEA-12P-200G	0.05497	94.50%	0.65	26
NSGA-II-20P-100G	0.13747	86.25%	0.65	26
MOEA/D-20P-100G	0.16394	83.61%	0.45	18
MOPSO-20P-100G	0.24083	75.92%	0.65	26
SPEA2-20P-100G	0.12993	87.01%	0.55	22
MOLEA-20P-100G	0.05622	94.38%	0.725	29
NSGA-II-40P-100G	0.10490	89.51%	0.65	26
MOEA/D-40P-100G	0.15864	84.14%	0.45	18
MOPSO-40P-100G	0.15655	84.35%	0.65	26
SPEA2-40P-100G	0.15599	84.40%	0.5	20
MOLEA-40P-100G	0.03992	96.01%	0.7	28

Note: The gray shadows mark algorithms with maximum coverage solutions under the same evolutionary generation and population size.

From Figure 10 and Table 9, the following two conclusions can be drawn: First, the MOLEA outperforms the four comparative algorithms in terms of the solution distribution and acquisition solution with the maximum coverage. It denotes that applying the training results for the Hubei to the imaging task planning problem in the Congo (K) is valid. Second, although the MOLEA outperforms the comparison algorithms, the maximum coverage rate was only 92.67% when the population size was 12 and the evolutionary generation was 100, and the maximum coverage rate was only 96.01% when the population size was 40 and the evolutionary generation was 100. There is still a 2% to 7% gap with the planning result in Hubei. This phenomenon could be attributed to the following two reasons: first, the search space for the optimal solution increased exponentially with the gradual increase in the scale of decision variables. This leads to an exponential decrease in the optimization efficiency of the evolutionary algorithm, which belongs to the essential characteristics of MOEA. Second, the characteristics of populations with different decision variable scales may differ. This led to a decrease in MOLEA performance when the training results for the Hubei were applied in the Congo (K) for testing. Therefore, in this experiment, we can only say that for a training result, the MOLEA can achieve superior results when the scale of decision variables changes within a specific range. However, it is difficult to say that MOLEA has high scalability in most cases, which requires further study.

4.8. Display of the Area Coverage

In this experiment, we visualized the mission planning results of the MOLEA and the comparison algorithms for Hubei and Congo (K). For planning results in the experiments above, the solution with the highest coverage was selected for each algorithm in each region, as displayed in Figures 11 and 12.

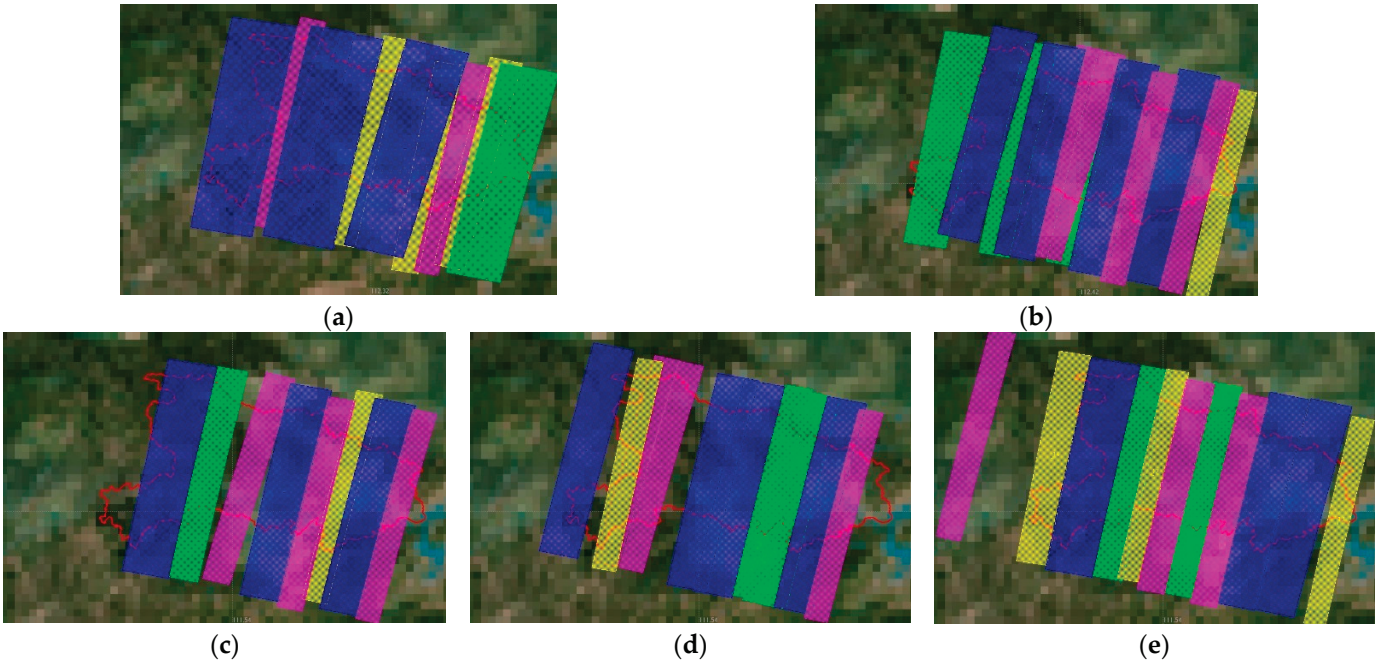


Figure 11. Hubei coverage results. (**a**) MOLEA-12P-100G (**b**) NSGA-II-12P-1000G (**c**) MOEA/D-12P-1000G (**d**) MOPSO-12P-1000G (**e**) SPEA2-12P-1000G.

In Figures 11 and 12, the green, blue, pink, and yellow strips represent the imaging strips of GF-1, GF-6, ZY1-02C, and ZY3, respectively. In Figure 11, the MOLEA, NSGA-II, and SPEA2 can achieve higher coverage than other algorithms, mutually verified in Table 6. However, SPEA2 exhibits a redundant strip, and the MOLEA requires fewer strips than NSGA-II. In Figure 12, the MOLEA and NSGA-II exhibit good coverage quality, the MOLEA can fully cover the main part of Congo (K), NSGA-II exhibits a small gap in the main body, and other comparison algorithms exhibit apparent gaps. Although no

algorithm entirely covered the protrusion in Congo (K), the MOLEA still had more strips to image the protrusion area than other algorithms. Therefore, the MOLEA still exhibited the best performance in the coverage display.

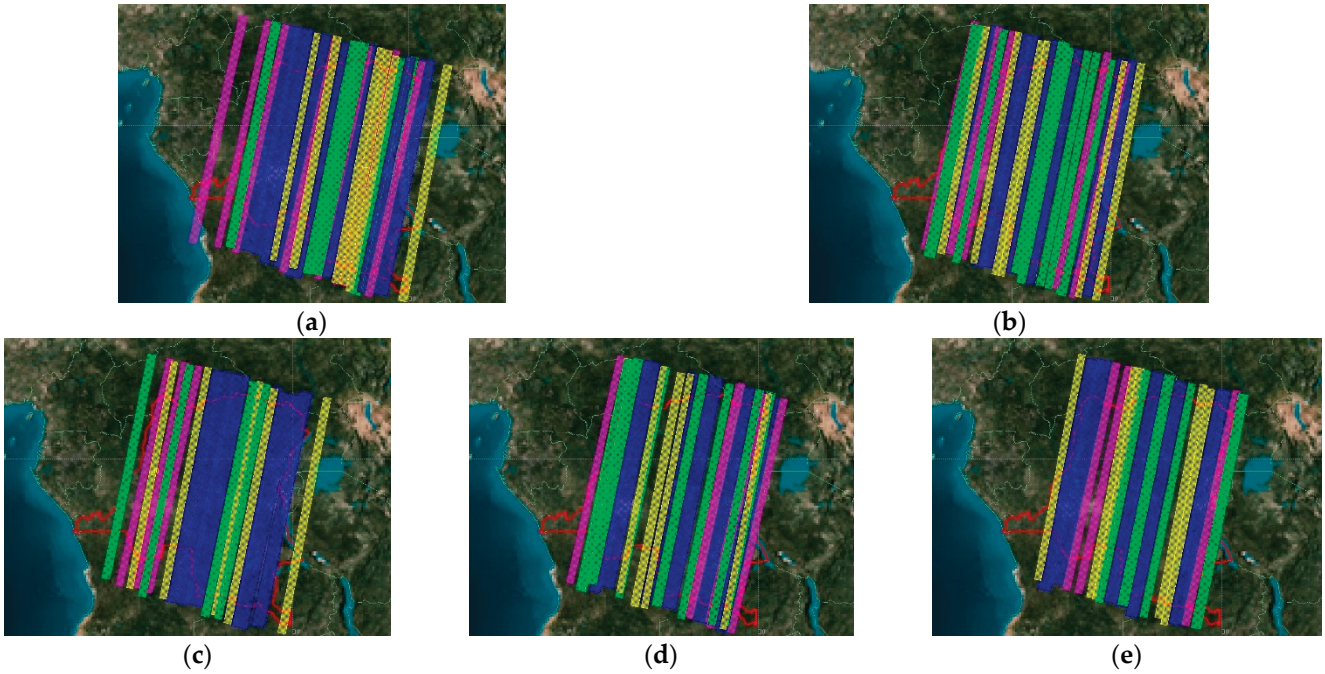


Figure 12. Congo (K) Coverage results. (**a**) MOLEA-40P-100G. (**b**) NSGA-II-40P-100G. (**c**) MOEA/D-40P-100G. (**d**) MOPSO-40P-100G. (**e**) SPEA2-40P-100G.

The regional visualization results of different algorithms are different, which is determined by the decision variables and objective functions of the multi-satellite imaging mission planning model. Different decision variables and objective function values can be obtained by using different algorithms to solve the model. In Figures 11 and 12, whether each strip is imaged and its imaging position are determined by decision variables. The total number of imaging strips and coverage area of each region are determined by the objective functions. The unreasonable distribution of strips and redundant strips are because the MOEA has not obtained the optimal solution to the problem.

5. Discussion

Imaging satellite mission planning scientifically arranges satellites for imaging based on user needs, maximizing the comprehensive imaging benefits. With the improvement of satellite quantity and quality, imaging satellite task planning technology can efficiently utilize satellite resources and accelerate the acquisition of regional remote sensing images for the production of regional surveying products. Mathematical modeling and optimization algorithm solving are key technologies for imaging satellite mission planning. Our previous work investigated the modeling problem of multi-satellite regional mapping. This article studies the optimization algorithm for solving multi-satellite regional mapping problems from the perspective of parameter configuration. Parameter configuration is an important factor affecting the performance of optimization algorithms, which has been emphasized in many studies [13,16,43]. To solve the problem of artificial and sub-optimal parameter configuration of existing optimization algorithms, we use DRL to solve it because DRL has good perception and decision-making ability. This paper uses deep learning to perceive the ever-changing population state and reinforcement learning to make the best evolutionary decisions among various populations. By this way, the optimization performance of optimization algorithms has been fully developed to achieve the highest area coverage in task planning problems with the least satellite resources. Satellite resources are efficiently utilized. In terms of runtime, the evolutionary efficiency of optimization algorithms with optimal parameters

is significantly higher than that of optimization algorithms with sub-optimal parameters. Therefore, optimization algorithms with optimal parameters obtain the optimal solution with fewer evolutionary generations, saving computational costs and runtime costs.

There are many optimization problems in the real world. The proposed MOLEA is not only applicable to imaging satellite task planning for regional mapping, but also to imaging satellite task planning problems for other imaging needs, such as fastest imaging time [44], stereo imaging [45,46], multi-region imaging [47], agile satellite imaging [48], etc. It can also be used for optimization problems in other fields, such as deep learning network optimization [49,50], traffic optimization [51], ship lock scheduling [52] and path planning [53], etc. An optimization algorithm with optimal parameter configuration can greatly save computational costs and economic costs of practical application problems.

In future work, we will mainly focus on two points: In terms of algorithms, we will study large-scale multi-objective optimization algorithms to solve large-area mapping problems, because the search space of solutions increases exponential growth with the increase in decision variables. In terms of application, the regional mapping in this paper refers to the regional orthophoto images, and the regional stereo images and multi-region imaging will be the focus of future research.

6. Conclusions

This paper has proposed a multi-objective learning evolutionary algorithm MOLEA based on DRL to solve the regional mission planning problem of imaging satellites. In the MOLEA, the DRL was used to simultaneously configure multiple evolutionary parameters according to the population state to maximize the cumulative benefit, considerably improving the convergence efficiency of the MOLEA. The proposed MOLEA effectively improves the training speed by replacing the practical application problem with the improved multi-objective functions as the training function. To the best of our knowledge, the MOLEA first proposed considering HV as the evolutionary reward of the multi-objective optimization problem. It achieves the ingenious combination of MOEA and DRL. Experimental results revealed that the proposed autonomous parameter configuration method could effectively improve the convergence speed of the algorithm in solving application problems. For the mission planning in Hubei, the planning solution obtained by MOLEA evolution for 100 generations was better than that obtained by the comparison algorithms evolution for 1000 generations. In terms of running time, the MOLEA improved by more than five times.

The proposed MOLEA is particularly suitable for application problems with high requirements for optimal solutions. The parameter configuration method studied in this paper is not limited to a multi-objective evolutionary algorithm, which can provide a reference for MOEA based on genetic algorithm, differential evolutionary, and particle swarm optimization. However, the parameter configuration of MOEAs is still an important issue that is easily overlooked, and the MOEAs based on DRL are still in the nascent stage of development. In the proposed MOLEA, the parameters' autonomous decision configuration based on DRL is only limited to small-scale problems, and the performance of training data is degraded when testing problems have different scales. In the future, we will study more stable MOEAs based on DRL to solve larger-scale application problems. Moreover, the proposed MOLEA can be used as a theoretical basis for future multi-objective optimization algorithms and satellite imaging mission planning research.

Author Contributions: Conceptualization, Y.C., X.S. and G.Z.; methodology, Y.C. and X.S.; validation, Y.C., X.S. and Z.L.; formal analysis, X.S.; investigation, Y.C.; resources, Y.C.; data curation, X.S.; writing—original draft preparation, Y.C.; writing—review and editing, Y.C.; visualization, Z.L.; supervision, X.S.; project administration, G.Z.; funding acquisition, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China (2022YFB3902803) and the National Natural Science Foundation of China under Grant 42171341.

Data Availability Statement: All data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, D.; Tian, H.; Zhou, G.; Ge, H. Regional mapping of human settlements in southeastern China with multi-sensor remotely sensed data. *Remote Sens. Environ.* **2008**, *112*, 3668–3679. [CrossRef]
2. Li, D.; Wang, Y.; Shao, Z. Informatization Surveying and Mapping in the New Geographic Information Age. *J. Wuhan Univ. (Inf. Sci. Ed.)* **2012**, *37*, 1–6+134.
3. Govekar, P.D.; Griffin, C.; Beggs, H. Multi-Sensor Sea Surface Temperature Products from the Australian Bureau of Meteorology. *Remote Sens.* **2022**, *14*, 3785. [CrossRef]
4. Li, T.; Hu, D.; Wang, Y.; Di, Y.; Liu, M. Correcting remote-sensed shaded image with urban surface radiative transfer model. *Int. J. Appl. Earth Obs.* **2022**, *106*, 102654. [CrossRef]
5. Yin, H.; Weng, L.; Li, Y.; Xia, M.; Hu, K.; Lin, H.; Qian, M. Attention-guided siamese networks for change detection in high-resolution remote sensing images. *Int. J. Appl. Earth Obs.* **2023**, *117*, 103206. [CrossRef]
6. Xuan, J.; Xin, Z.; Liao, G.; Huang, P.; Wang, Z.; Sun, Y. Change Detection Based on Fusion Difference Image and Multi-Scale Morphological Reconstruction for SAR Images. *Remote Sens.* **2022**, *14*, 3604. [CrossRef]
7. Lu, Z.; Shen, X.; Li, D.; Li, D.; Chen, Y.; Wang, D.; Shen, S. Multiple super-agile satellite collaborative mission planning for area target imaging. *Int. J. Appl. Earth Obs.* **2023**, *117*, 103211. [CrossRef]
8. Sharma, S.; Kumar, V. A Comprehensive Review on Multi-objective Optimization Techniques: Past, Present and Future. *Arch. Comput. Methods Eng.* **2022**, *29*, 5605–5633. [CrossRef]
9. Tian, Y.; Cheng, R.; Zhang, X.; Jin, Y. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput. Intell. Mag.* **2017**, *12*, 73–87. [CrossRef]
10. Lin, Q.; Liu, S.; Zhu, Q.; Tang, C.; Song, R.; Chen, J.; Coello, C.A.C.; Wong, K.; Zhang, J. Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. *IEEE Trans. Evol. Comput.* **2016**, *22*, 32–46. [CrossRef]
11. Kropp, I.; Nejadhashemi, A.P.; Deb, K. Benefits of sparse population sampling in multi-objective evolutionary computing for large-Scale sparse optimization problems. *Swarm Evol. Comput.* **2022**, *69*, 101025. [CrossRef]
12. Karafotias, G.; Hoogendoorn, M.; Eiben, A.E. Parameter Control in Evolutionary Algorithms: Trends and Challenges. *IEEE Trans. Evol. Comput.* **2015**, *19*, 167–187. [CrossRef]
13. Eiben, A.E.; Smith, J. From evolutionary computation to the evolution of things. *Nature* **2015**, *521*, 476–482. [CrossRef]
14. Deng, W.; Zhang, X.; Zhou, Y.; Liu, Y.; Zhou, X.; Chen, H.; Zhao, H. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inform. Sci.* **2022**, *585*, 441–453. [CrossRef]
15. Huang, C.; Li, Y.; Yao, X. A survey of automatic parameter tuning methods for metaheuristics. *IEEE Trans. Evol. Comput.* **2019**, *24*, 201–216. [CrossRef]
16. Parpinelli, R.S.; Plichoski, G.F.; Silva, R.S.D.; Narloch, P.H. A review of techniques for online control of parameters in swarm intelligence and evolutionary computation algorithms. *Int. J. Bio-Inspir. Comput.* **2019**, *13*, 1–20. [CrossRef]
17. Li, S.; Shen, X.; Yao, H.; Zhang, G.; Liu, Y. An optimization method for the roll angle of circumlunar satellites for regional imaging missions. *J. Wuhan Univ. (Inf. Sci. Ed.)* **2019**, *44*, 593–600.
18. Ming, F.; Gong, W.; Yang, Y.; Liao, Z. Constrained multimodal multi-objective optimization: Test problem construction and algorithm design. *Swarm Evol. Comput.* **2023**, *76*, 101209. [CrossRef]
19. Dai, L.; Zhang, L.; Chen, Z.; Ding, W. Collaborative granular sieving: A deterministic multi-evolutionary algorithm for multimodal optimization problems. *Inform. Sci.* **2022**, *613*, 288–308. [CrossRef]
20. Xue, Y.; Tang, T.; Pang, W.; Liu, A.X. Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers. *Appl. Soft Comput.* **2020**, *88*, 106031. [CrossRef]
21. Yang, H.; Tao, S.; Zhang, Z.; Cai, Z.; Gao, S. Spatial information sampling: Another feedback mechanism of realizing adaptive parameter control in meta-heuristic algorithms. *Int. J. Bio-Inspir. Comput.* **2022**, *19*, 48–58. [CrossRef]
22. Bao, Q.; Wang, M.; Dai, G.; Chen, X.; Song, Z. Dynamical decomposition and selection based evolutionary algorithm for many-objective optimization. *Appl. Soft Comput.* **2023**, *141*, 110295. [CrossRef]
23. Xue, Y.; Cai, X.; Neri, F. A multi-objective evolutionary algorithm with interval based initialization and self-adaptive crossover operator for large-scale feature selection in classification. *Appl. Soft Comput.* **2022**, *127*, 109420. [CrossRef]
24. Eiben, A.E.; Schut, M.C.; de Wilde, A.R. Is Self-adaptation of Selection Pressure and Population Size Possible?—A Case Study. In *Parallel Problem Solving from Nature—PPSN IX, Proceedings of the International Conference on Parallel Problem Solving from Nature*; Runarsson, T.P., Beyer, H., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X., Eds.; Springer: Berlin/Heidelberg, Germany, 9 September 2006; pp. 900–909.
25. Botvinick, M.; Ritter, S.; Wang, J.X.; Kurth-Nelson, Z.; Blundell, C.; Hassabis, D. Reinforcement Learning, Fast and Slow. *Trends Cogn. Sci.* **2019**, *23*, 408–422. [CrossRef] [PubMed]
26. Eiben, A.E.; Michalewicz, Z.; Schoenauer, M.; Smith, J.E. Parameter control in evolutionary algorithms. In *Parameter Setting in Evolutionary Algorithms*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 19–46.

27. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
28. Schuchardt, J.; Golkov, V.; Cremers, D. Learning to evolve. *arXiv* **2019**, arXiv:1905.03389.
29. Li, K.; Zhang, T.; Wang, R. Deep Reinforcement Learning for Multiobjective Optimization. *IEEE Trans. Cybern.* **2021**, *51*, 3103–3114. [CrossRef] [PubMed]
30. Song, Y.; Wei, L.; Yang, Q.; Wu, J.; Xing, L.; Chen, Y. RL-GA: A Reinforcement Learning-based Genetic Algorithm for Electromagnetic Detection Satellite Scheduling Problem. *Swarm Evol. Comput.* **2023**, *77*, 101236. [CrossRef]
31. Tian, Y.; Li, X.; Ma, H.; Zhang, X.; Tan, K.C.; Jin, Y. Deep Reinforcement Learning Based Adaptive Operator Selection for Evolutionary Multi-Objective Optimization. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**, *7*, 1051–1064. [CrossRef]
32. Reijnen, R.; Zhang, Y.; Bukhsh, Z.; Guzek, M. Deep Reinforcement Learning for Adaptive Parameter Control in Differential Evolution for Multi-Objective Optimization. In Proceedings of the 2022 IEEE Symposium Series on Computational Intelligence (SSCI), Singapore, 4–7 December 2022; pp. 804–811.
33. Li, T.; Meng, Y.; Tang, L. Scheduling of Continuous Annealing With a Multi-Objective Differential Evolution Algorithm Based on Deep Reinforcement Learning. *IEEE Trans. Autom. Sci. Eng.* 2023; *Early Access*.
34. Chen, Y.; Xu, M.; Shen, X.; Zhang, G.; Lu, Z.; Xu, J. A Multi-Objective Modeling Method of Multi-Satellite Imaging Task Planning for Large Regional Mapping. *Remote Sens.* **2020**, *12*, 344. [CrossRef]
35. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
36. Hedar, A.R. Global Optimization Test Problems. 2016. Available online: http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm (accessed on 15 January 2020).
37. Surjanovic, S.; Bingham, D. Virtual Library of Simulation Experiments: Test Functions and Datasets. 2017. Available online: https://www.sfu.ca/ssurjano/optimization.html (accessed on 15 January 2020).
38. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv* **2015**, arXiv:1506.02438.
39. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
40. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
41. Mofid, H.; Jazayeri-Rad, H.; Shahbazian, M.; Fetanat, A. Enhancing the performance of a parallel nitrogen expansion liquefaction process (NELP) using the multi-objective particle swarm optimization (MOPSO) algorithm. *Energy* **2019**, *172*, 286–303. [CrossRef]
42. Wahid, A.; Gao, X.; Andreae, P. Multi-objective clustering ensemble for high-dimensional data based on Strength Pareto Evolutionary Algorithm (SPEA-II). In Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, France, 19–21 October 2015; pp. 1–9.
43. Muzid, S. An Adaptive Approach to Controlling Parameters of Evolutionary Algorithms. *J. Phys. Conf. Ser.* **2020**, *1430*, 12048. [CrossRef]
44. Haiquan, S.U.N.; Wei, X.I.A.; Xiaoxuan, H.U.; Chongyan, X.U. Earth observation satellite scheduling for emergency tasks. *J. Syst. Eng. Electron.* **2019**, *30*, 931–945.
45. Cui, K.; Xiang, J.; Zhang, Y. Mission planning optimization of video satellite for ground multi-object staring imaging. *Adv. Space Res.* **2018**, *61*, 1476–1489. [CrossRef]
46. Kim, J.; Ahn, J.; Choi, H.; Cho, D. Task Scheduling of Agile Satellites with Transition Time and Stereoscopic Imaging Constraints. *J. Aerosp. Inf. Syst.* **2020**, *17*, 285–293. [CrossRef]
47. Lu, Z.; Shen, X.; Li, D.; Chen, Y. Integrated Imaging Mission Planning Modeling Method for Multi-Type Targets for Super-Agile Earth Observation Satellite. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 4156–4169. [CrossRef]
48. Qiu, W.; Xu, C.; Ren, Z.; Teo, K.L. Scheduling and Planning Framework for Time Delay Integration Imaging by Agile Satellite. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 189–205. [CrossRef]
49. Young, S.R.; Rose, D.C.; Karnowski, T.P.; Lim, S.; Patton, R.M. Optimizing deep learning hyper-parameters through an evolutionary algorithm. In Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments; Association for Computing Machinery, Austin, TX, USA, 15 November 2015.
50. Martín, A.; Lara-Cabrera, R.; Fuentes-Hurtado, F.; Naranjo, V.; Camacho, D. EvoDeep: A new evolutionary approach for automatic Deep Neural Networks parametrization. *J. Parallel Distr. Com.* **2018**, *117*, 180–191. [CrossRef]
51. Ma, W.; Wan, L.; Yu, C.; Zou, L.; Zheng, J. Multi-objective optimization of traffic signals based on vehicle trajectory data at isolated intersections. *Transp. Res. Part C Emerg. Technol.* **2020**, *120*, 102821. [CrossRef]
52. Ji, B.; Zhang, D.; Yu, S.S.; Zhang, B. Optimally solving the generalized serial-lock scheduling problem from a graph-theory-based multi-commodity network perspective. *Eur. J. Oper. Res.* **2021**, *288*, 47–62. [CrossRef]
53. Yang, X.; Yang, Y.; Li, Y.; Yang, X. Path planning for guided passengers during evacuation in subway station based on multi-objective optimization. *Appl. Math. Model.* **2022**, *111*, 777–801. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

Gated Recurrent Unit Embedded with Dual Spatial Convolution for Long-Term Traffic Flow Prediction

Qingyong Zhang †, Lingfeng Zhou †, Yixin Su, Huiwen Xia * and Bingrong Xu

School of Automation, Wuhan University of Technology, 122 Luoshi Road, Wuhan 430070, China; qyzhang@whut.edu.cn (Q.Z.); zlf19990924@whut.edu.cn (L.Z.); suyixin@whut.edu.cn (Y.S.); bingrongxu@whut.edu.cn (B.X.)

* Correspondence: wendyxia@whut.edu.cn

† These authors contributed equally to this work.

Abstract: Considering the spatial and temporal correlation of traffic flow data is essential to improve the accuracy of traffic flow prediction. This paper proposes a traffic flow prediction model named Dual Spatial Convolution Gated Recurrent Unit (DSC-GRU). In particular, the GRU is embedded with the DSC unit to enable the model to synchronously capture the spatiotemporal dependence. When considering spatial correlation, current prediction models consider only nearest-neighbor spatial features and ignore or simply overlay global spatial features. The DSC unit models the adjacent spatial dependence by the traditional static graph and the global spatial dependence through a novel dependency graph, which is generated by calculating the correlation between nodes based on the correlation coefficient. More than that, the DSC unit quantifies the different contributions of the adjacent and global spatial correlation with a modified gated mechanism. Experimental results based on two real-world datasets show that the DSC-GRU model can effectively capture the spatiotemporal dependence of traffic data. The prediction precision is better than the baseline and state-of-the-art models.

Keywords: traffic flow prediction; graph convolution network; gated mechanism; recurrent neural network; spatiotemporal dependence

1. Introduction

AS people's standards of living improve, their desire for a better life becomes increasingly strong. The convenience of commuting and traveling attracts more and more families to choose cars as their mode of transportation. In most cities across the country, road occupancy rate continues to increase. while the average driving speed during rush peak hours has sustained a decrease. This leads to the deterioration of urban road operations. In this case, not only the efficiency of urban road use will be significantly reduced but also fuel cannot be completely burned when cars drive at a low speed. This shortens the service life of the engine and produces a lot of harmful gas pollution, both of which increase the cost of car maintenance and environmental management [1]. As an integral component of an Intelligent Transportation System (ITS), how to improve the accuracy of traffic flow prediction and use it as the basis for rational route planning to reduce road congestion has emerged as a research focal point [2].

In order to enhance the precision of traffic flow prediction, many scholars in the field of traffic flow prediction have proposed various prediction models in recent years. These models for predicting traffic flow can be generally categorized into three groups, including statistical analysis models, machine learning models, and deep learning models. Statistical analysis models commonly used in traffic flow prediction include the Historical Average model(HA) [3], the Autoregressive Integrated Moving Average model(ARIMA) [4], and the Kalman filtering model [5]. These models can achieve better results when dealing with

static data. However, traffic flow data change dynamically over time and have strong spatial dependence. Statistical analysis methods are insensitive to these types of data; so, these models often fail to achieve the desired results when performing traffic flow prediction tasks [6]. Machine learning models for traffic flow prediction are introduced to solve the above problem, such as the K-Nearest Neighbor model (KNN) [7], the support vector regression model [8], and the Bayesian network model [9]. For the sake of achieving superior predictions from machine learning models, it is necessary to set appropriate parameters for the models based on a large amount of a priori knowledge. It is difficult and laborious to set the optimal hyperparameters of the model artificially; so, the power of the model is often not fully exploited. As artificial intelligence is increasingly applied to various fields, deep learning is receiving more and more attention. Deep-learning-based traffic flow prediction models stack multiple layers of neural network to capture various dependence of the traffic data. The effect of hyperparameters is weakened [10], and the influence of human factors is reduced [11]. For example, recurrent neural networks (RNNs) [12] and their variant long short-term memory (LSTM) [13], as well as Gated Recurrent Units (GRUs) [14], have achieved great results in performing temporal features extraction [15]. Graph convolutional networks (GCNs) [16], which evolved from convolutional neural networks (CNNs) [17], are commonly used for learning the temporal features of traffic data.

The main challenges of traffic prediction task originate from the spatial and temporal dependencies inherent in traffic flow data [18]:

(1) Spatial dependence: The layout of the nodes in the urban road network is generally an irregular graphical topological structure, as shown as Figure 1.

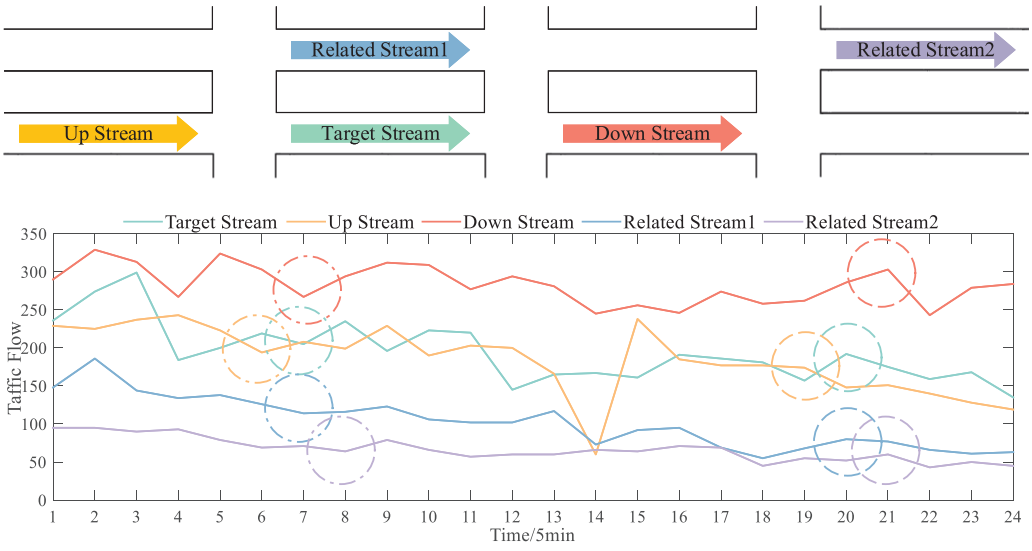


Figure 1. Effect of the topological structure. Similar trends exist between nodes.

Such a spatial structure will make the traffic state between the nodes interfere with each other. The traffic state of upstream and downstream nodes in the same segment has different effects on the target node [19]. It is also found that even if the node is far from the target node in the road network space, it will affect the prediction of future traffic flow at the target node [20]. As shown in Figure 1, it can be observed that certain segments of the time series share similar trends, as indicated by the circled areas. Additionally, both upstream and downstream traffic flow data exhibit similar patterns, albeit with a delay in the downstream traffic flow changes. This suggests a correlation between the two types of traffic flow and a potential lag effect in the downstream traffic flow. Moreover, these nodes with high spatial correlation have some kind of correlational relationship or common influencing factors that lead to changes in traffic status with similar trends (e.g., Related Stream). How to exploit the existing relationship between spatial nodes is important to improve the accuracy of traffic flow prediction.

(2) Temporal dependence: Temporal modeling of traffic flow data requires consideration of the inherent properties of traffic flows.

The traffic flow data of each node in the urban road network vary dynamically and periodically with time [21]. Meanwhile, traffic flow data at urban road network nodes can also be subject to many uncertainties (e.g., weather, traffic accidents, etc.) [22].

Figure 2, where ①, ②, ③, and ④ represent the traffic flow data of one workday, weekend, holiday, and week, respectively, demonstrates that the traffic flow data have the following characteristics in the time dimension. First, traffic flow data have obvious daily periodicity (e.g., ① and ②) and weekly periodicity (e.g., ③), and the change pattern of weekdays is significantly different from that of weekends. Second, since the change pattern of traffic data on holidays is different from that of weekdays, it will break the periodicity of traffic state change (e.g., ④). To meet the requirement of prediction accuracy, the above dynamic change patterns must be fully considered.

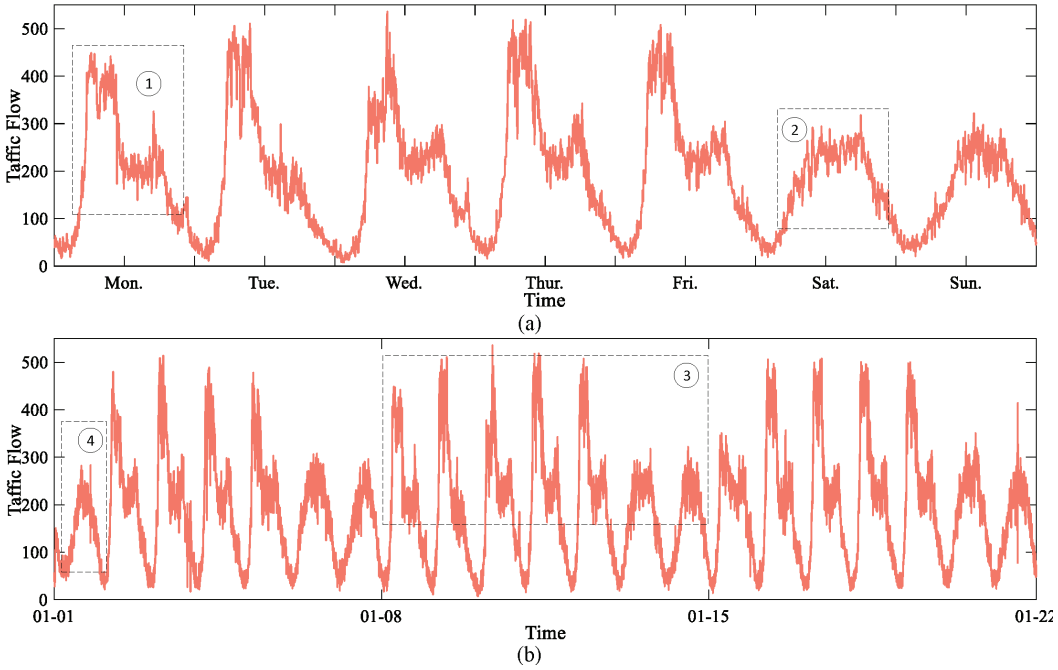


Figure 2. Periodicity of traffic flow data: (**a**) Daily periodicity. (**b**) Weekly periodicity. ① traffic flow data of a workday. ② traffic flow data of a weekday. ③ traffic flow data of a holiday. ④ traffic flow data of one week.

To address the challenges in traffic flow prediction, the main contributions of this work are as follows:

(1) We propose a new traffic flow prediction model, termed Dual Spatial Convolution Gated Recurrent Unit (DSC-GRU), which considers the spatiotemporal dependence of the traffic flow data and achieves a more accurate prediction result.

(2) Based on the R-Square (R^2), we design a correlation matrix that reflects whether the nodes are spatially related or not, allowing the graph convolution network to be aware of global features, and develop a gated-mechanism-based spatial graph convolution module that integrates the neighborhood and global features through a learnable parameter matrix.

(3) The proposed model is capable of synchronously incorporating both temporal and spatial correlations of traffic flow data. This capability is achieved by embedding a Dual Spatial Convolution (DSC) module into the GRU architecture. The DSC facilitates the integration of spatial information into the state of the GRU without compromising its ability to learn the complex temporal dynamics of the traffic flow data.

Our model is evaluated on two real-world traffic datasets. Compared with the baseline and state-of-art models, the results show that the DSC-GRU outperforms all of them.

The rest of this paper is organized as follows. Section 2 presents related work and introduces various existing models used for traffic flow prediction; Section 3 is the problem description, which defines the traffic flow prediction problem; Section 4 is the methodology, which presents the details of the proposed model; Section 5 is the experimental setting, which presents the relevant details of the experiments; Section 6 is the experimental results and analysis, and it visualizes the results; and Section 7 concludes the whole paper.

2. Related Work

Facing the high-dimensional temporal data of traffic flow, the performance of traditional statistical analysis models and machine learning models are increasingly unable to meet the accuracy requirement of traffic flow prediction. Meanwhile, with the improvement of information collection technology and computer processing power, many researchers have turned their attention to deep-learning-based traffic flow prediction methods [23].

Deep learning methods commonly used in traffic flow prediction can be broadly classified into two categories: temporal feature extraction models and spatiotemporal feature extraction models.

(1) Temporal feature extraction model:

A fundamental feature of time series data is that data at historical moments will affect the pattern of data changes at future moments. Wang et al. [24] designed a traffic flow prediction model based on back propagation neural network. The predicted value is obtained after a series of nonlinear variations in the input data. Huang et al. [25] proposed the deep belief network model, consisting of a deep belief network for unsupervised traffic flow feature learning and a multitasks regression layer for supervised prediction. Lv et al. [26] used a stacked autoencoder to learn traffic flow features, and it was trained bottom-up layer by layer with the greedy wolf algorithm. Logistic regression was used to complete the traffic flow prediction task. These models ignore the sequential relationship in the time series. The output corresponding to the input of each moment is not affected by the previous input. As a result, the prediction accuracy of these models is not satisfactory. To consider the sequential relationship of time series data, RNNs have been proposed. As a classical model for time series processing, RNNs can handle the sequential relationship in the time series. However, they have some defects, such as gradient disappearance and gradient explosion. Many improvements on RNN models have overcome the above problems and achieved better results in time series prediction, such as Ma et al. [27], who were the first to apply LSTM to traffic flow prediction. The error decay problem in the backpropagation process has been overcome. Sun et al. [28] stacked multiple GRUs to enhance the ability to extract the temporal features. In addition, the Seq2Seq model provided a novel idea for traffic flow prediction. The Seq2Seq model was first proposed to solve the machine translation problem [29]. Sutskever et al. [30] created an end-to-end sequence learning method based on the Seq2Seq model. The encoder and decoder are formed by stacking multilayer LSTMs. The encoder encodes the input sequence into a vector of fixed dimensions, and the decoder subsequently decodes the vector to obtain the target sequence.

(2) Spatiotemporal feature extraction model:

With the iterative upgrade of deep learning techniques, researchers increasingly feel that simple time series prediction models cannot meet the accuracy requirement of traffic flow prediction task. Since the topological structure of the road network affects the traffic state of the nodes, researchers have begun to turn their attention to prediction models with the fusion of spatiotemporal features [31]. CNNs point the way to learning spatial features for traffic flow prediction models [32]. The ConvLSTM model created by Shi et al. [33] combined a CNN with LSTM for the first time to address the prediction problem. The CNN can handle the graph structure of Euclidean space well, but it is hard for it to process the graph structure of the road network, which belongs to non-Euclidean space. The GCN was raised to map the traffic road network to Euclidean space for processing. Zhao et al. [34] designed a Temporal Graph Convolutional Network (T-GCN) which used the GCN model

to learn the complex spatial topological structure in the road network space and used GRU to obtain the time dependence of the traffic data. The attention mechanism is also widely used in traffic flow prediction tasks. Guo et al. [35] proposed the attention-based spatiotemporal graph convolutional network. The attention mechanism is introduced to model three types of temporal dependence. Much research has been conducted to overcome the drawback that the GCN can only consider the information of adjacent nodes and give equal weight to neighboring nodes. Huang et al. [36] developed a diffusion convolutional recurrent neural network model to integrate the information of all nodes within n-hops by means of random wandering. Bao et al. [37] designed a spatial–temporal complex graph convolution network which considers the influence of external factors such as weather and facilities around the nodes. Wu et al. [38] raised the Graph WaveNet. The adaptive dependency matrix is used to capture the temporal dependence accurately. Song et al. [39] designed a spatiotemporal synchronous modeling mechanism to effectively capture the complex localized spatiotemporal dependence. The improvements on the road network graph structure have become the mainstream methods to capture the spatial dependence of traffic data.

Motivated by the above method, the R^2 correlation matrix is designed to learn the global spatial dependence of the traffic data, which enhances the spatiotemporal feature capturing ability of our proposed DSC-GRU model.

3. Problem Description

The traffic flow prediction task aims at predicting the traffic state of the urban road network at a future moment using historical traffic flow data [40]. Traffic flow data includes vehicle speed, traffic flow, and road network density [41]. In this study, we use traffic flow as an example to elaborate the model.

Traffic road network: The road network $G = (V, E)$ is used to describe the spatial topology structure of the urban road network. $V = \{v_1, v_2, \cdots, v_N\}$ denotes the set of road network nodes, and N is the number of nodes. Since the graph used in our model is an undirected graph, E is the set of edges connected by any two road network nodes.

Feature matrix: The traffic flow data $X \in R^{N \times L}$, where L represents the length of the time series collected at each node; then, the traffic flow data at the time t is $X_t \in R^{N \times 1}$. Based on the above definitions, the traffic sequences input to the model and the predicted sequences output from the model are given as $X_{in} = \{X_{t-h+1}, X_{t-h+2}, \cdots, X_t\} \in R^{N \times h}$ and $X_{out} = \{X_t, X_{t1}, \cdots, X_{t+p}\} \in R^{N \times p}$, respectively, where h denotes the length of the historical time series used for prediction, and p represents the size of the predicting window.

Mapping function: The traffic flow prediction task can be described as a process of training a mapping function f from input to output, which takes historical traffic flow data X_{in} with known graph structures G as input and calculates the traffic flow values in the following p moments, as shown in Equation (1).

$$X_{out} = f(X_{in}; G) \tag{1}$$

4. Methodology

This section explains how the DSC-GRU model can consider spatiotemporal features and complete the traffic prediction task.

4.1. Overview

The DSC-GRU model consists of a GRU embedded with a DSC. The framework of our proposed model is shown in Figure 3, where B, H, and P represent the batch size of data used to train the prediction model once, the length of the input model's historical data, and the length of the model's output prediction data, respectively. And sampling refers to dividing the feature matrix into a number of H-length matrix blocks along the L dimension and randomly selecting the B-matrix blocks as the input data to the model for a training epoch.

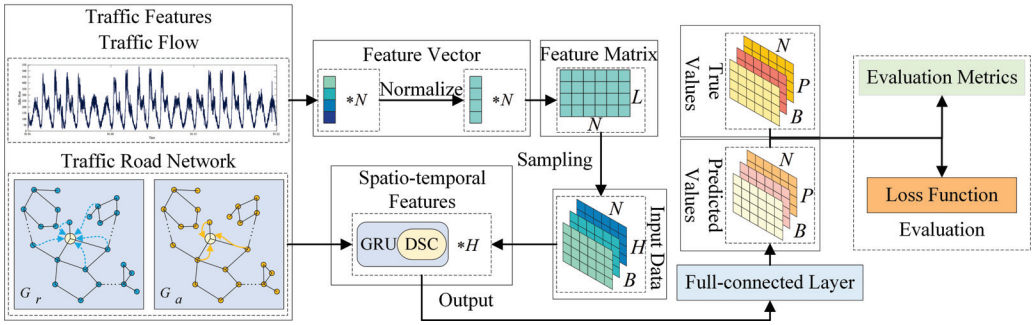


Figure 3. Framework of the DSC-GRU model.

Based on the existing graph structure G_a and G_r, the DSG-GRU model is used to extract the spatial and temporal features of the traffic flow data. The DSC-GRU model retains the ability of the GRU to extract dynamic time series features. The embedded DSC is used to obtain the spatial topological structure of the urban road network, thus capturing the spatial characteristics of the traffic data. Finally, the output prediction results are obtained through a fully connected layer.

The core ideas of the DSC-GRU can be summarized into three aspects: (1) Spatial correlation matrix: design a correlation matrix based on the correlation coefficient to learn the spatial correlation of the traffic road network. (2) Spatial features fusion: a gated mechanism is introduced to integrate adjacent features and global features of the traffic road network to optimize the prediction performance of the network. (3) Temporal features extraction: adopt the GRU network to capture the temporal correlation of the traffic flow data.

4.2. Spatial Dependence Modeling

Obtaining the spatial features of urban road network is a critical problem that has long plagued traffic flow prediction. Although CNN models can capture spatial features in Euclidean space well, the complex topological structure of urban road network belongs to non-Euclidean space, so CNNs cannot accurately capture the spatial dependencies of traffic data [42]. In recent years, the GCN model, which evolved from CNNs, has been widely used in computer vision [43], biochemistry [44], and other fields with good results due to its great ability to handle various graph structures.

4.2.1. Graph Data Definition

In order to more accurately capture the spatial dependencies in traffic flow data, the GCN model maps the spatial topology of the traffic road network from a non-Euclidean space to a Euclidean space. However, our proposed model involves two distinct scales of spatial structures; so, it is necessary to define the spatial structure graph data separately. Figure 4a,b show the distribution of aggregation nodes in the adjacency graph and the global graph, respectively.

The adjacency graph, which is mapped to Euclidean space to obtain the adjacency matrix $A_a \in R^{N\times N}$, characterizes the spatial structure of adjacent nodes. The adjacency matrix denotes that the connection relationship in the set of edges E is defined in Equation (2),

$$A_{a_{ij}} = \begin{cases} 1\left(e_{ij} \in E\right) \\ 0\left(e_{ij} \notin E\right) \end{cases} \tag{2}$$

where $A_{a_{ij}} = 1$ represents node i connected to node j and $A_{a_{ij}} = 0$ vice versa.

The adjacency matrix A_a used in traditional graph convolution can generally only aggregate the spatial structure within the adjacent nodes or n-hop nodes of the road network. These simple graphs are based on the assumption that the upstream and downstream traffic states on the same road section have a strong influence on the traffic state of the target node, which neglect the topological structure of the whole urban road network.

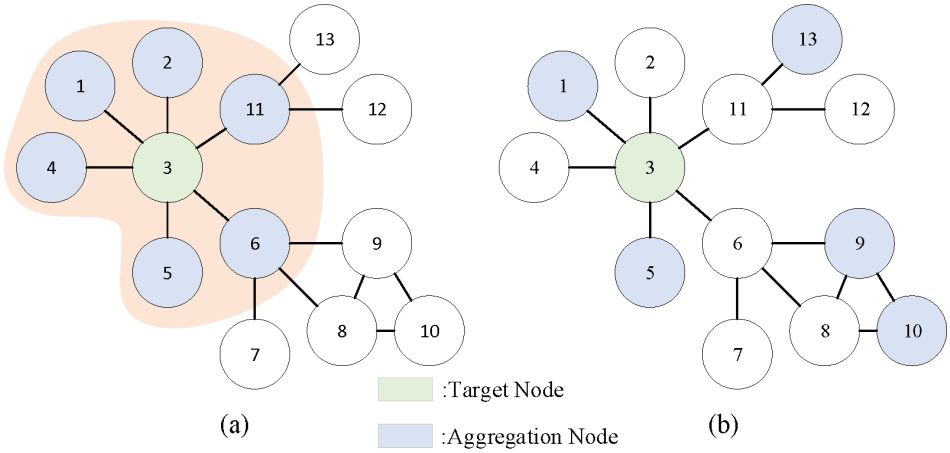


Figure 4. Node distribution: (**a**) Adjacent space. (**b**) Global space.

In order to obtain the spatial feature of the whole road network, some studies used the KNN [45] algorithm to calculate the correlation between the two time series so that the topological structure of the urban road network can be fully explored. The KNN algorithm selects K nodes with the highest relevance to the target node in the entire road network. It may happen that nodes are not actually strongly correlated but are identified as strongly correlated by the KNN algorithm. From such nodes, the model will learn a lot of useless information.

To make spatial information aggregation more flexible and accurate, the correlation coefficient R^2 is introduced to quantify the correlation between different nodes. The R^2 is derived as follows:

$$SST = \sum_{k=0}^{L} (X_i^k - \overline{X_i})^2 \tag{3}$$

$$SSR = \sum_{k=0}^{L} (X_j^k - \overline{X_i})^2 \tag{4}$$

$$SSE = \sum_{k=0}^{L} \left(X_i^k - X_j^k\right)^2 \tag{5}$$

$$R_{ij}^2 = 1 - \frac{SSE}{SST} = \frac{SSR}{SST} \tag{6}$$

where SST, SSR, and SSE refer to Sum of Squares Total, Sum of Squares Regression, and Sum of Squares Error, respectively, R_{ij}^2 represents the correlation between the node i and the node j, the value closer to 1 means the greater correlation, and X_i^k and $\overline{X_i}$ represent the traffic flow value and the average traffic flow value of the node i at the time k, respectively. The relationship between the R^2 correlation coefficient and correlation strength is shown in Table 1 below.

Table 1. The relationship between the R^2 correlation coefficient and correlation strength.

The Value Range of R^2	Correlation Strength
$R^2 < 0.2$	None
$0.2 \sim 0.4$	Weak
$0.4 \sim 0.6$	Moderate
$0.6 \sim 0.8$	Strong
$0.8 \sim 1.0$	Extreme

The correlation of road network nodes is calculated based on the R_{ij}^2 correlation coefficient. It is assumed that if the correlation coefficient exceeds the threshold, then the

corresponding nodes are considered to be strongly correlated. On this basis, A_r is defined as Equation (7),

$$A_{r_j} = \begin{cases} 1(R_j \geq threshold) \\ 0(R_j < threshold) \end{cases} \tag{7}$$

where $A_{r_{ij}} = 1$ means that the corresponding node is highly correlated and $A_{a_{ij}} = 0$ vice versa.

4.2.2. Graph Data Processing

The graph structure used in most traffic flow prediction task is often an undirected graph based on the urban road network. In this case, the GCN network integrates the characteristics of the nodes and their neighbors based on the weight matrix W and the adjacency matrix A. The GCN network contains a two-layer structure, which is calculated as shown below:

$$GCN(A, X) = softmax\Big(AReLU(AXW^{(0)})W^{(1)}\Big) \tag{8}$$

where $W^{(0)}$ and $W^{(1)}$ are the weight matrices of the first and second layer GCN network, respectively.

Adjacency matrix $\tilde{A}$ is defined to merge the node's own information,

$$\tilde{A} = A + \lambda I_N \tag{9}$$

where I_N is an identity matrix with the same dimension as the number of nodes, and λ is the weight factor and takes a value of 1, indicating that the information of this node is as important as that of its neighbors.

The GCN model normalizes the rows and columns of the adjacency matrix $\tilde{A}$ by the symmetric normalized Laplacian matrix,

$$A^{sys} = D^{-1/2}\tilde{A}D^{-1/2} \tag{10}$$

where the degree matrix $D \in R^{N \times N}$ is the diagonal matrix, D_{ii} is the element on the degree matrix, and a_{ii} represents the element of the i row and j column of the matrix A.

$$D_{ii} = \sum_{j=1}^{N} a_{ij} \tag{11}$$

In summary, the standard format of the GCN network can be described as the following equation:

$$GCN(A^{\text{sys}}, X) = softmax\Big(A^{\text{sys}}ReLU(A^{\text{sys}}XW^{(0)})W^{(1)}\Big) \tag{12}$$

By replacing the adjacency matrix A in the traditional GCN network with the adjacent matrix A_a and correlation matrix A_r, the topological structure information of the adjacent space and global space can be integrated.

4.2.3. Dual Spatial Convolution

The process of integrating the structural information of the adjacent space and the global space is shown in Figure 5a,b, respectively.

If the adjacent and global information are simply superimposed, it must be assumed that their importance is equal. However, it is impossible to artificially judge the significance of the adjacent and the global information in the traffic flow prediction task. A simple superposition will seriously reduce the prediction accuracy of the model. The role of the gated mechanism is that the learning of the weight matrix can adjust the importance of

the adjacent information and the global information to the target node to achieve better prediction accuracy.

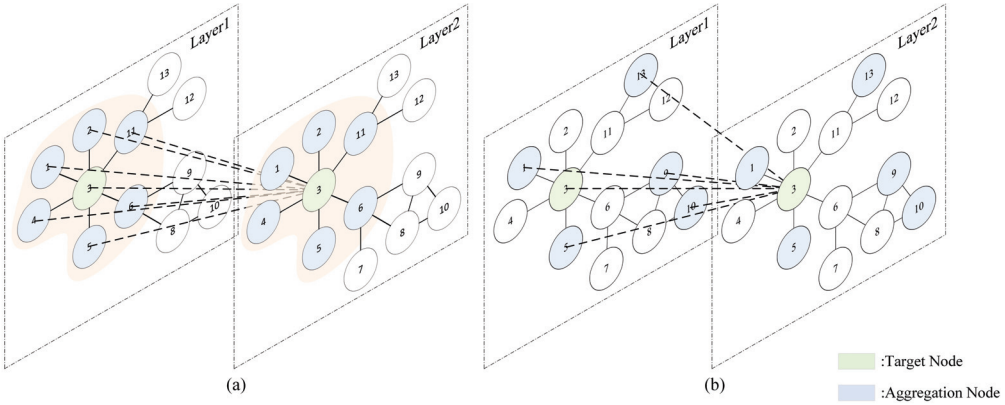


Figure 5. Integration process: (**a**) Adjacent space. (**b**) Global space.

In addition, there is a particular case if the calculated R^2 correlation between the target node and its neighbor exceeds the set threshold. The information of those neighbor nodes will also be fused when integrating global information. To some extent, it alleviates the problem that the traditional graph convolution model ignores the differences in the influence of different neighborhoods on the target node.

The output of the graph convolution operation for G_a and G_r is recorded as GCN_a and GCN_r, respectively. They are used as the inputs to the gated unit. The overall structure of the DSC is shown in Figure 6, where "$\oplus$" and "$\odot$" represent the addition and multiplication of the corresponding elements of the isotype matrix, respectively, and "1−" means 1 minus the elements of each position of the matrix. $\sigma(\cdot)$ represents the sigmoid activation function, which maps the input to the [0, 1] interval. Its forward propagation equations are shown as Equations (13)–(16):

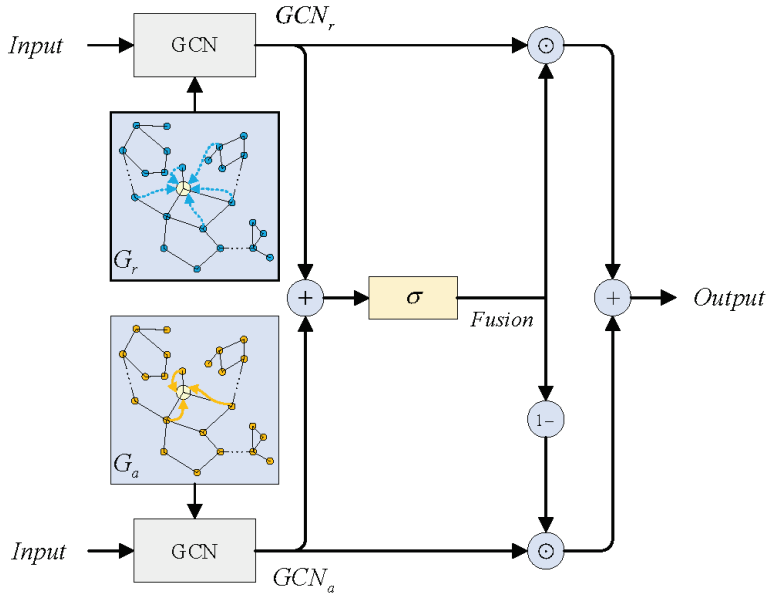


Figure 6. The overall structure of the DSC unit.

$$GCN_a = softmax\left(A_a^{sys} ReLU(A_a^{sys} X W_a^{(0)}) W_a^{(1)}\right) \tag{13}$$

$$GCN_r = softmax\left(A_r^{sys} ReLU(A_r^{sys} X W_r^{(0)}) W_r^{(1)}\right) \tag{14}$$

$$Fusion = \sigma\left(W_o \cdot (GCN_a + GCN_r) + b_o\right) \tag{15}$$

$$Ouput = GCN_a * (1 - Fusion) + GCN_r * Fusion \tag{16}$$

where W and b are the parameter matrix, $Output$ represents the output of the DSC unit, and $Fusion$ incorporates the spatial information of the adjacent space and the global space. $Fusion$ is used to adjust the proportion of the importance of the global information. The closer to 1, the higher the importance of the global information. Then, there is more focus on the spatial information of the whole road network space when performing the prediction task and, conversely, focus on the spatial information of the adjacent space.

In summary, the DSC can balance the relative importance relationship between the adjacent space and the global space through the gated mechanism. This allows the DSC to better extract the topological structure of the urban road network and capture the spatial dependence of traffic data.

4.3. Spatiotemporal Dependence Modeling

In the previous subsection, we used the DSC to extract the spatial dependence of traffic data. Still, another critical issue in the traffic flow prediction task is extracting time series features. The most commonly used method for processing time series data is the RNN, and it achieves promising results on many models. Due to the defects of gradient explosion and gradient disappearance, traditional RNNs cannot obtain the temporal features of traffic flow data, which has long-term time-dependent data. The LSTM and the GRU are proposed as variants of the RNN. They provide solutions to the above problems. The principles of the LSTM and the GRU are similar [46]. Both are based on RNNs and add a gating mechanism and long-term information memory cells, thus solving the problem of long-term dependence on time series.

Compared with LSTM, GRUs have fewer parameters, require less computational power, and are easier to converge during the training process with less risk of overfitting. Therefore, the spatiotemporal dependence is modeled based on the GRU framework in this paper.

In order to obtain both the temporal-dependent and spatial-dependent traffic flow data, the DSC-GRU traffic flow prediction model is proposed by embedding the DSC into the framework of GRU. The overall prediction process of the DSC-GRU model is shown in Figure 7, where H_t represents the hidden layer state output by the DSC-GRU cell at the time step t, and Y_{out} is the final result of the DSC-GRU cell output. r_t, u_t, and c_t are the calculation results of the reset gate, the update gate, and the candidate hidden state at the time step t, respectively. The left side is the schematic diagram of the forward propagation of the model, and the right side is the specific structure of the DSC-GRU.

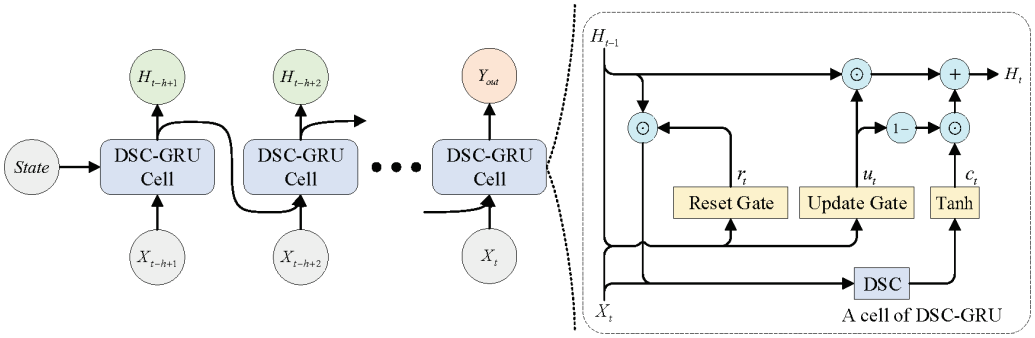


Figure 7. The overall prediction process of the DSC-GRU model.

The DSC-GRU model forward propagation is formulated as Equations (17)∼(20):

$$r_t = \sigma(W_r \cdot [X_r, H_{t-1}] + b_r) \tag{17}$$

$$u_t = \sigma(W_u \cdot [X_r, H_{t-1}] + b_u) \tag{18}$$

$$c_t = tanh\big(W_c \cdot DSC([X_t, (r_t * H_{t-1})]) + b_c\big) \tag{19}$$

$$H_t = u_t * H_{t-1} + (1 - u_t) * c_t \tag{20}$$

where W, b is the parameter matrix of each state in the training process, $\sigma(\cdot)$ has the same meaning as the previous section, $tanh(\cdot)$ represents the tanh activation function, which maps the input to the $[-1, 1]$ interval to prevent the gradient explosion due to the large number of parameters in the backpropagation process, and $DSC(\cdot)$ denotes the result obtained by DSC cell calculation.

The DSC-GRU model retains the ability of the GRU model to capture the long-term dependence of the time series by processing the input X_t at the moment t and the hidden state H_{t-1} at the previous moment. The temporal information of the current moment and the historical moment are fused. The dynamic characteristics of the time series are preserved. On this basis, the DSC unit is embedded to capture the spatial dependence of the traffic data. The validity fraction of the historical spatiotemporal information is obtained through reset gates and passed to the next moment. On the moment t, the input hidden layer state of the DSC-GRU model already contains both temporal and spatial information from the previous moment. The DSC fuses valid historical spatiotemporal information with the input of the current moment. On the one hand, the spatial features of the current input data are fused. On the other hand, the importance of spatial features in historical information is strengthened. Up to this point, the embedding of the DSC has added the spatial information of the urban road network to each state in the GRU model. This spatial information is passed from the previous time step to the next time step as the GRU model passes temporal information. Throughout the forward propagation process, the above operation is repeated several times to continuously reinforce the learning of spatial features by the DSC to improve the model's ability to extract spatial dependencies. The pseudocode of the model is shown in Appendix A.

In summary, the DSC-GRU model can explore the hidden spatial dependence and temporal dependence of traffic flow data. For one thing, the DSC unit is used to obtain the topological structure of the whole urban road network and then obtain the spatial dependence of the traffic flow data. For another, the GRU model is used to obtain the temporal dependence of traffic flow data by considering the long-term temporal relationship in traffic flow data. Finally, the task of traffic flow prediction is completed.

5. Experiment Setting

In order to verify the effectiveness and generalization ability of the traffic flow prediction model proposed in this paper. The prediction performance of the DSC-GRU model is evaluated based on two real datasets, and the results are visualized.

5.1. Data Description

In this paper, the PeMS04 and PeMS08 datasets are used to evaluate model performance. Both of them are highway datasets collected by the California Department of Transportation based on the Caltrans Performance Measurement System (PeMS). The system collects real-time traffic status at 30 s intervals and aggregates all information collected at 5 min intervals. All datasets contain three characteristics dimensions: vehicle speed, traffic flow, and network density. In this paper, the traffic flow is used as the research object. The training set, validation set, and test set are divided in the ratio of 6:2:2. The differences between PeMS04 and PeMS08 are as follows:

(1) PeMS04: This dataset contains a total of 307 road nodes. It was collected for 59 days from 1 January 2018 to 28 February 2018, of which 35 days were used as the training set, 12 days as the validation set, and 12 days as the test set.

(2) PeMS08: This dataset contains a total of 170 road nodes. It was collected for 62 days from 1 July 2016 to 31 August 2016, of which 38 days were used as the training set, 12 days as the validation set, and 12 days as the test set.

5.2. Evaluation Metrics

The predictive performance of our DSC-GRU model is evaluated by the following four evaluation metrics.

(1) Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{L}\left|y_i^j - \tilde{y}_i^j\right| \tag{21}$$

(2) Root Mean Squared Error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{L}\left(y_i^j - \tilde{y}_i^j\right)^2} \tag{22}$$

(3) Mean Absolute Percentage Error (MAPE):

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{L}\frac{\left|y_i^j - \tilde{y}_i^j\right|}{\left|y_i^j\right| + \epsilon} \tag{23}$$

(4) Coefficient of determination (R^2):

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{L}\left(y_i^j - \tilde{y}_i^j\right)^2}{\sum_{i=1}^{N}\sum_{j=1}^{L}\left(y_i^j - \overline{y}\right)^2} \tag{24}$$

where y_i^j and $\tilde{y}_i^j$ represent the real traffic state and predicted traffic state of node i at the time j, respectively, ε is a subsidiary term, and $\overline{y}$ is the average value of traffic flow data. MAE and RMSE are used to measure the degree of deviation of the model's predicted value from the true value. MAPE indicates the relative size of the deviation of the model's predicted value from the true value. R^2 weighs the ability of the model's predicted value to represent the true value. The smaller the MAE, RMSE, and MAPE and the larger the R^2, the better the prediction performance of the model.

5.3. Loss Function

The goal of training the traffic flow prediction model is to minimize the deviation of the predicted value from the true value of the traffic flow data, i.e., to minimize the value of the loss function. The loss function used to train this model is shown in Equation (25),

$$Loss = \begin{cases} 0.5(y-\widehat{y})^2/beta, & if|y-\widehat{y}| < beta \\ |y-\widehat{y}| - 0.5 * beta, & otherwise \end{cases} \tag{25}$$

where y and $\hat{y}$ represent the true value and predicted value of traffic flow, respectively. The default value for *beta* takes 1.

The advantage of this type of loss function is that the curve is relatively smooth. When the difference between the predicted value and the true value is slight, the gradient will not be too small. When the difference is significant, it is not easy for gradient explosion to appear.

5.4. Parameter Setting

The parameters of the DSC-GRU model include batch size, learning rate, number of iterations, R^2 threshold, historical data length h, prediction length p, and the number of neurons in the DSC and the GRU hidden layer (dim_{DSC_h} and dim_{GRU_h}). In the experimental

process, the batch size is set to 64, the learning rate is set to 0.002, the number of iterations is set 2000, and the h and p are set to 12, i.e., the historical information of the past hour is used to predict the traffic flow of the future hour. Furthermore, R^2 threshold, dim_{DSC_h}, and dim_{GRU_h} are set to 0.8, 512, and 128, respectively.

In addition, a variable learning rate mechanism is introduced for better learning of traffic flow data. When the current optimal value does not appear in 10 consecutive training epochs, the learning rate of the model is considered too large, and the network hovers around the optimal value. The learning rate is reduced to the original 0.1 so that the model can learn more carefully. When the current optimal value appears, the learning rate is reset to 0.002 to make the model converge quickly in the new solution space. And random seeds are set in the same group of experiments to ensure the same initialization parameters of the model and reduce the effect of random errors.

5.5. Compared Methods

In order to verify the validity and the advancement of the proposed model, the model is compared with five baseline models considering a single feature of the traffic flow data as well as two spatiotemporal models proposed in recent years. These methods are as follows:

(1) HA [3]: The average value of historical traffic flow data is used as the traffic flow prediction for the next moment.

(2) ARIMA [4]: Traffic flow data are considered as a time series with seasonal patterns. It is possible to separate the signal from the noise and obtain predicted values by extrapolating the signal to the future.

(3) LSTM [27]: One of the variants of the recurrent neural network, which obtains long-term dependencies in the time dimension by introducing a gated mechanism. It can solve the problem of gradient disappearance and gradient explosion in traditional recurrent neural networks. (4) GRU [28]: The principle is similar to that of LSTM, except that the number of parameters in the training process is smaller and the computation is simpler.

(5) GCN [16]: Mapping the road network space from non-Euclidean space to Euclidean space. The spatial features of traffic data are extracted from neighboring nodes in the traffic road network.

(6) T-GCN [34]: Combines the advantages of both the GCN and the GRU, which consider the temporal dependence and spatial dependence of traffic data.

(7) Graph WaveNet [38]: The adaptive dependency matrix and the dilated convolution structure are designed to accurately capture the spatiotemporal dependencies in traffic flow data.

6. Experiment Result and Analysis

This section tests the prediction performance of the DSC-GRU model by setting comparison experiments, different R^2 thresholds, structural parameters, and ablation experiments.

6.1. Comparison Experiments

The performance of the different models is shown in Table 2, where the bolded parts are the models proposed in this paper. It is shown that the DSC-GRU model outperformed all comparison models on the four evaluation metrics.

The experiment results are analyzed in four aspects, as follows. Initially, the reason why the statistical analysis models (HA and ARIMA) perform worse than the deep learning models is that these models are insensitive to the dynamically changing traffic flow data and cannot explore the spatial information of the traffic data. Secondly, either the deep learning models that consider only temporal dependence (LSTM and GRU) or those that consider only spatial dependence (GCN) do not achieve satisfactory prediction results. The reason is that the traffic data are essentially time series data, which are influenced by the topological structure of the road network. Neglecting either the temporal or the spatial dependence will lead to a decrease in prediction accuracy. Thirdly, the spatiotemporal characteristics models

(T-GCN and Graph WaveNet) have significantly better prediction performance than the LSTM, GRU, and GCN. The improvement in the prediction performance results from these models taking the spatiotemporal dependence of traffic flow data into consideration rather than single dependence. This demonstrates that the spatiotemporal dependence of the traffic data is of great importance for improving prediction accuracy. Finally, the DSC-GRU model proposed in this paper also considers the spatiotemporal dependence of traffic flow data simultaneously. As opposed to the above spatiotemporal models that only consider the spatial characteristics of nodes within a certain range, the DSC-GRU model can capture the global spatial characteristics of nodes. Therefore, the prediction performance is better than that of the T-GCN and Graph WaveNet models. Specifically, compared with the T-GCN and Graph WaveNet, the MAE value is decreased by 10.03% and 6.68%, the MAPE value is lowered by 13.34% and 5.52% on the PeMS04 dataset, the MAE value is reduced by 13.02% and 12.03%, and the MAPE value is dropped by 13.53% and 9.68% on the PeMS08 dataset, respectively.

Table 2. The performance comparison of different models.

Models	PeMS04				PeMS08			
	MAE	**RMSE**	**MAPE**	R^2	**MAE**	**RMSE**	**MAPE**	R^2
HA	38.09	54.51	28.38%	0.880	32.14	46.06	20.36%	0.900
ARIMA	35.19	50.05	24.18%	0.889	29.12	41.95	18.32%	0.912
LSTM	25.64	39.13	16.49%	0.937	19.50	30.16	11.45%	0.956
GRU	24.35	37.84	15.50%	0.942	18.97	29.96	11.19%	0.957
GCN	34.98	51.45	23.17%	0.894	28.97	41.55	17.49%	0.923
T-GCN	21.74	34.28	14.02%	0.951	18.05	26.93	10.57%	0.964
Graph WaveNet	20.96	33.94	13.76%	0.956	17.87	26.63	10.12%	0.967
DSC-GRU	**19.56**	**31.44**	**12.15%**	**0.960**	**15.72**	**25.10**	**9.14%**	**0.970**

Figures 8 and 9 show the trends of each evaluation metric for different deep learning models based on the PeMS04 and PeMS08 datasets with different horizons. The DSC-GRU model outperforms the the comparison models in all horizons, and the curve rises slowly with the horizon. The experimental results show that our proposed model has better stability.

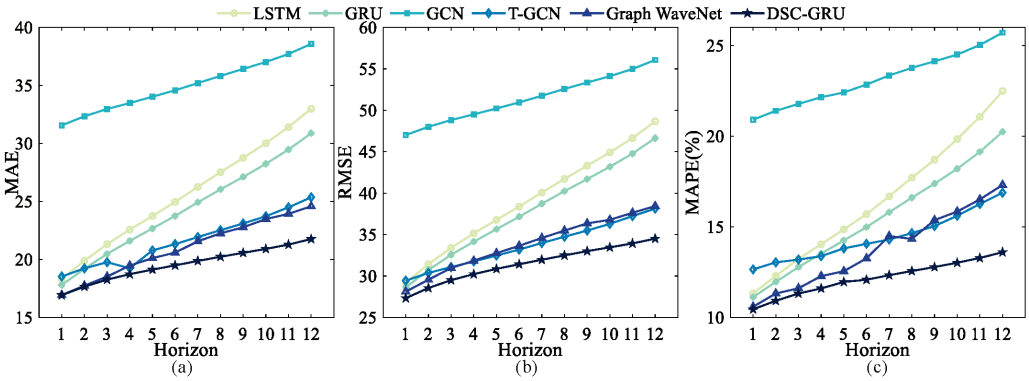


Figure 8. Evaluation metrics at different horizons of the model on PeMS04: (**a**) MAE. (**b**) RMSE. (**c**) MAPE.

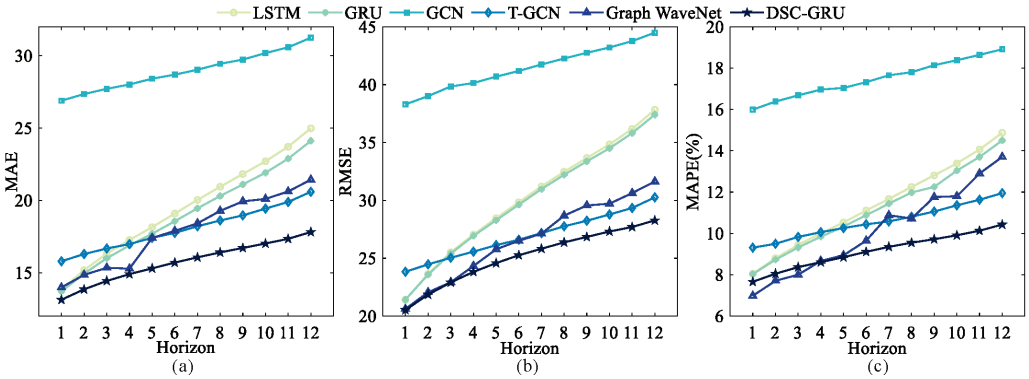


Figure 9. Evaluation metrics at different horizons of the model on PeMS08: (**a**) MAE. (**b**) RMSE. (**c**) MAPE.

To show the prediction effect of the model more intuitively, the real value and predicted value of the traffic flow data of node 9 for one day and one week in the PeMS04 and PeMS08 datasets are visualized, and the results are shown in Figure 10.

From the visualization results in Figure 10, it can be seen that the model prediction values always follow the real traffic values regardless of all the horizons of the model prediction. This indicates that the DSC-GRU model can correctly consider the spatial and temporal information of the traffic flow data and make predictions.

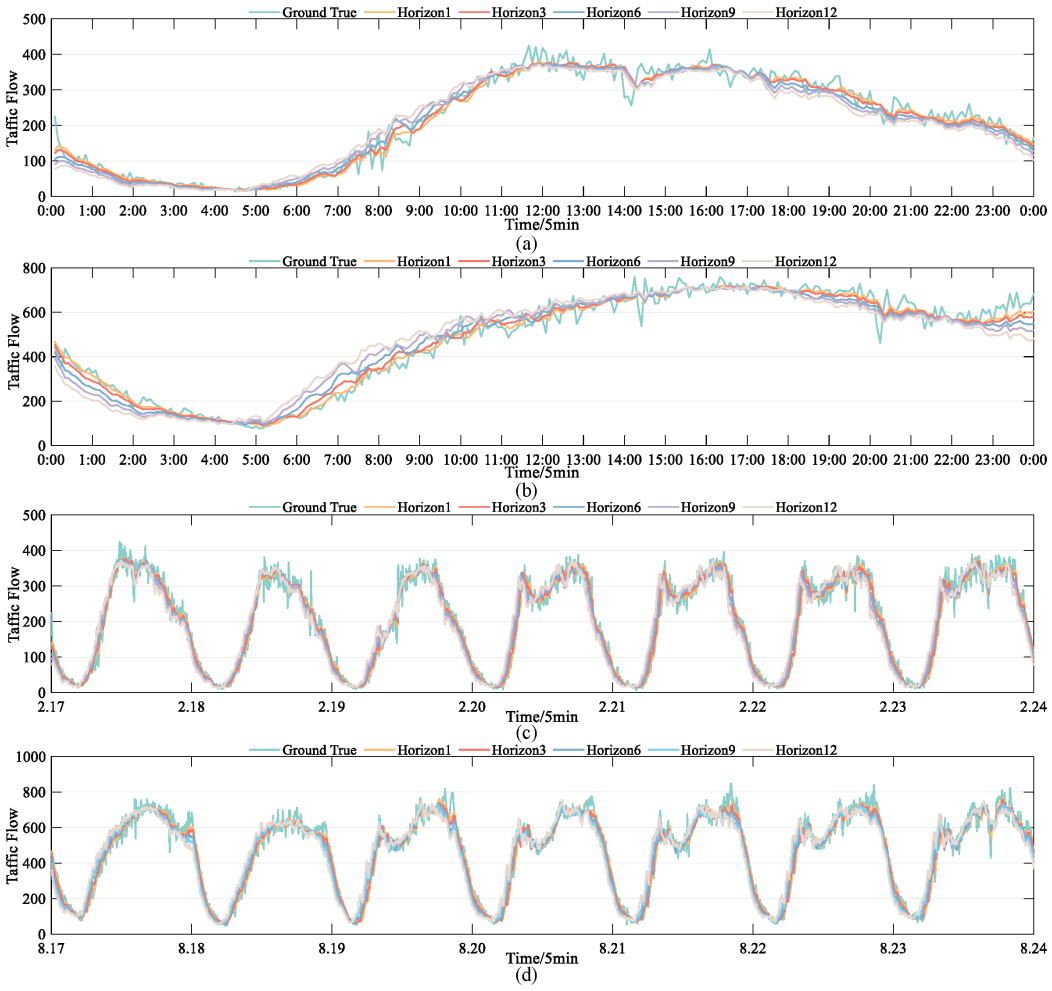


Figure 10. Visualization of prediction results: (**a**) One-day traffic flow at node 9 on PeMS04. (**b**) One-day traffic flow at node 9 on PeMS08. (**c**) One-week traffic flow at node 9 on PeMS04. (**d**) One-week traffic flow at node 9 on PeMS08.

6.2. Effect of Hyperparameters

This section discusses the role played by three different hyperparameters in the model. The effect of hyperparameters on the prediction performance of the model is tested by altering one of the hyperparameters while ensuring that the other hyperparameters remain constant.

6.2.1. Effect of R^2 Threshold Value

According to Equation (6), the node correlations in the PeMS04 and PeMS08 datasets are calculated. The correlations from node 101 to 120 are visualized and shown in Figure 11.

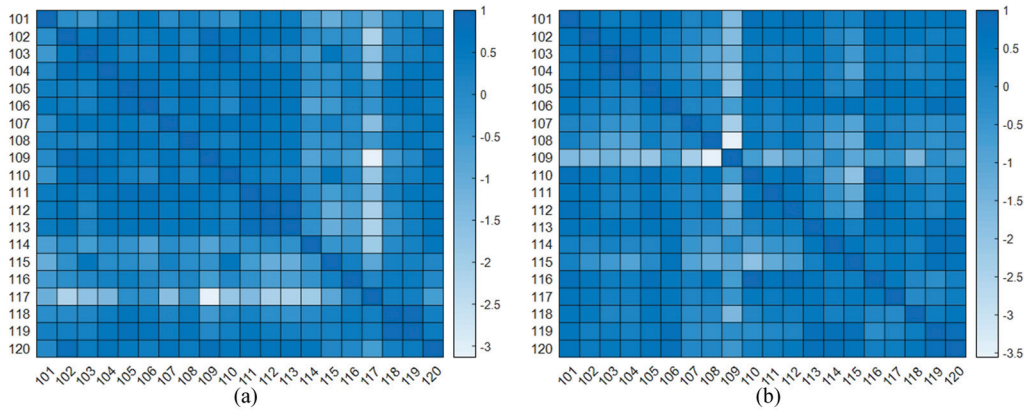


Figure 11. Partial node correlation heat map: (**a**) Heatmap for PeMS04. (**b**) Heatmap for PeMS08.

From the figure above, it can be seen that the magnitude of the correlation between nodes can be effectively quantified by the R^2 correlation coefficient. The larger value indicates that the traffic flow state of the corresponding two nodes is more similar. By setting the threshold value, the nodes with similar traffic flow characteristics to the target node are screened out to provide the basis for the model to aggregate the global spatial information of the road network.

Based on the above analysis, the R^2 thresholds affect the information integration of the global space of the road network. They are set to [0.6, 0.7, 0.8, 0.9] according to the case of strong correlation only. By setting different thresholds, the proposed model can control how many nodes are fused to the target node when the global spatial information of the road network is integrated. The larger the threshold, the higher the learning worth of these spatial nodes and the less invalid node information aggregated. The smaller the threshold, the more node information is aggregated, and more spatial information can be obtained. Therefore, it is worth investigating how to set the threshold value so that the model can aggregate the high-quality node information as much as possible while obtaining more spatial information. Therefore, this section tests the model by setting different thresholds to find a balance between high-quality information and more spatial information. dim_{DSC_h} and dim_{GRU_h} are set to 512 and 128, respectively. The experimental results under different threshold values are shown in Figure 12.

Comparing the experimental results based on the two datasets, it is found that the change trends of the evaluation metrics are not consistent when the threshold value is changed. This is reasonable, because PeMS04 and PeMS08 collect traffic state from different regions. The different spatial characteristics information from the different regions will lead to differences in the model when aggregating spatial information. In addition, the model proposed in this paper has a better ability to fit the traffic flow data when the R^2 threshold is set to 0.8.

The experimental results show that in both datasets, setting the R^2 threshold too high or too low will lead to a decrease in the predictive ability of the model. This is due to the fact that the larger the threshold is set, the less information is aggregated to the target nodes, which makes the model unable to capture the global spatial structure of the road

network well. Conversely, the smaller the threshold is set, the more invalid information is learned by the model, as more information is aggregated to the target node. When the threshold value is set moderately, the model can better balance the amount of aggregated information and the effectiveness of the information.

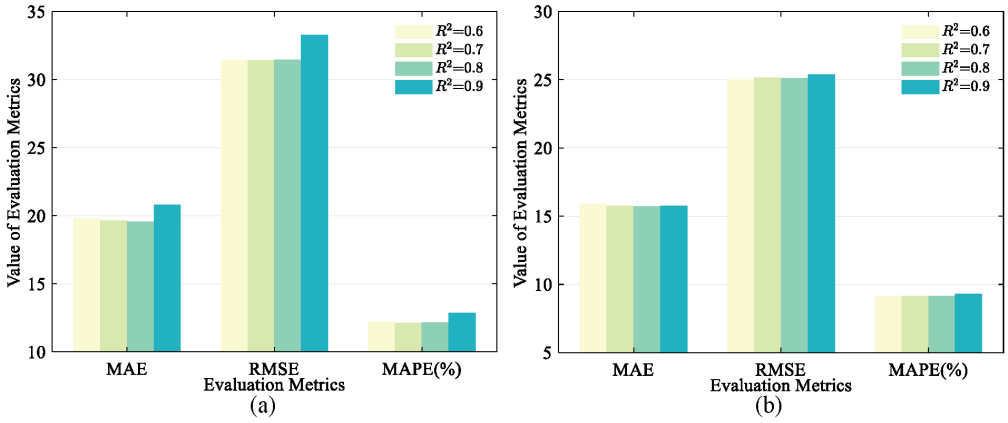


Figure 12. Evaluation metrics under different R^2 threshold values: (**a**) PeMS04. (**b**) PeMS08.

6.2.2. Effect of Model Structure Parameters

The number of neurons in the hidden layer will affect the fitting ability of the model. Too few neurons may lead to the insufficient ability of the model to extract features from traffic flow data. Too many neurons may increase the burden of the model and make the model training time costlier [47].

Considering that the GCN network in the DSC processes the entire input data, increasing the number of hidden layer neurons has little impact on the training time of the model; so, dim_{DSC_h} is set to [16, 32, 64, 128, 256, 512]. In addition, the R^2 threshold is set to 0.8 and dim_{GRU_h} is set to 128. The experimental results are shown in Figure 13.

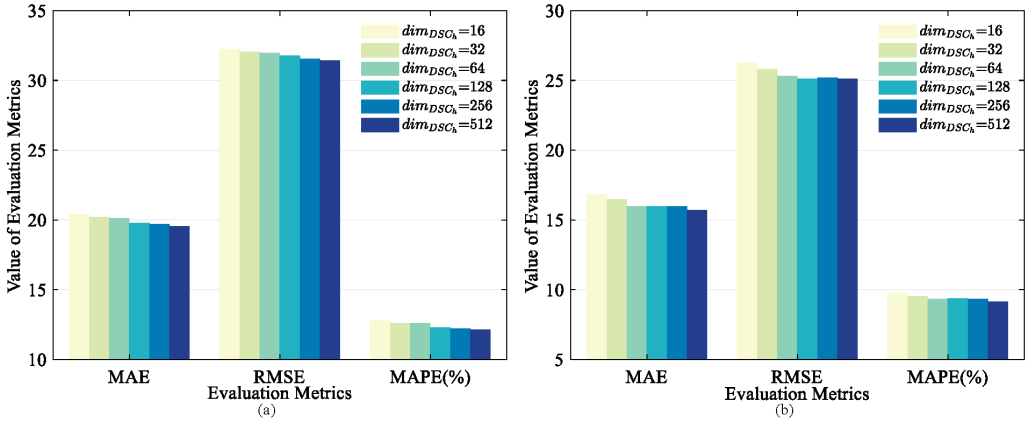


Figure 13. Evaluation metrics under different numbers of DSC hidden layer neurons: (**a**) PeMS04. (**b**) PeMS08.

Similarly, since the GRU must process the input at each time step, increasing the number of hidden layer neurons will lead to a significant increase in model training time; so, dim_{GRU_h} is set to [8, 16, 32, 64, 128, 256]. In addition, the R^2 threshold is set to 0.8 and dim_{DSC_h} is set to 512. The experimental results are shown in Figure 14.

According to the experimental results, we can draw the following conclusions. First of all, the DSC-GRU model is insensitive to the dim_{DSC_h}, but it does decrease slowly. The GCN in the DSC has the properties of local correlation and shared weight. These properties allow the DSC to capture structural information between nodes and have some robustness to changes in hidden layer neurons while maintaining the structure unchanged.

Simultaneously, as the dim_{GRU_h} increases, the prediction accuracy of the model initially improves rapidly. The GRU introduces gated mechanisms to control the flow of information, and the parameters of these gated units are learned through the network. When there is a change in the dim_{GRU_h}, it affects the input and output of the gated units, which in turn affects the flow and storage of information. Therefore, the GRU model is sensitive to the dim_{GRU_h}. After the dim_{GRU_h} reaches a certain number, the improvement of model accuracy by increasing the number of neurons becomes weak. This is because too large a dim_{GRU_h} may lead to overfitting, which limits performance gains. The cost of model training increases significantly as the number of neurons grows. The other parameters remain unchanged. When the dim_{DSC_h} is set to 512, the training time to an epoch is already twice as long as when the number of neurons is 16. Similarly, when the dim_{GRU_h} is set to 128, the training time to an epoch is already three times as long as that of 8 neurons. It is undeniable that increasing the number of neurons can improve the accuracy of the model. Still, it is meaningless to blindly increase the number of neurons wihtout considering the significant increase in training cost.

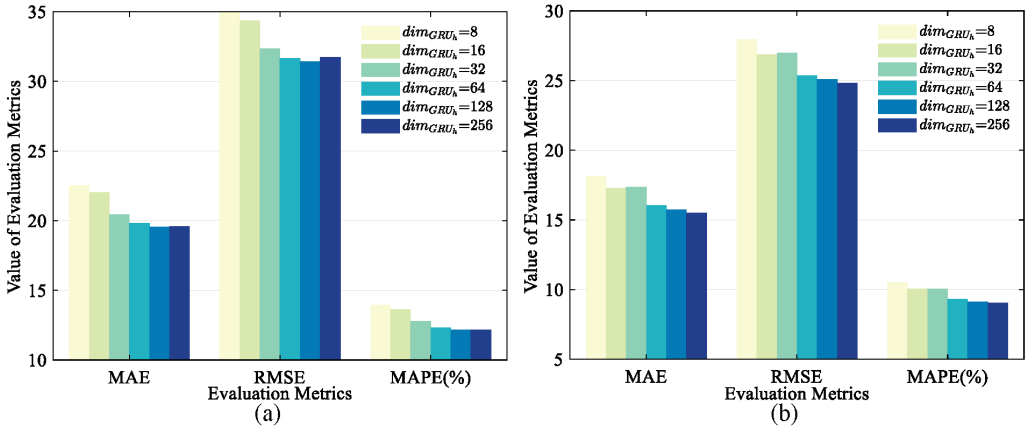


Figure 14. Evaluation metrics under different numbers of GRU hidden layer neurons: (**a**) PeMS04. (**b**) PeMS08.

6.3. Ablation Experiments

The DSC-GRU model consists of two components based on the gated mechanism: the dual spatial graph convolution and the temporal dependence recurrent neural network. To explore the performance of each component, the prediction performance is compared between the DSC-GRU model and its two components.

The ablation experiments are set up as follows. (1) DSC: this model ignores the time dependence in the traffic data and only considers the entire spatial topological structure of the road network. (2) GRU: the traditional GRU model considers the long-term temporal dependence of the traffic data but ignores the spatial topological structure of the road network. The threshold is set to 0.8, and the dim_{DSC_h} and dim_{GRU_h} are set to 512 and 128, respectively. The experimental results are shown in Table 3.

Table 3. The result of ablation experiments.

Models	PeMS04				PeMS08			
	MAE	**RMSE**	**MAPE**	R^2	**MAE**	**RMSE**	**MAPE**	R^2
DSC	27.13	40.58	17.95%	0.934	22.02	32.82	13.07%	0.949
GRU	24.35	37.84	15.50%	0.942	18.97	29.96	11.19%	0.957
DSC-GRU	19.56	31.44	12.15%	0.960	15.72	25.10	9.14%	0.970

The experiment results show that DSC has the worst prediction accuracy, because traffic flow data are essentially a kind of time series data. The potential time dependence in the

historical traffic flow data will affect the prediction results. The DSC model only considers the spatial dependence but ignores the time dependence, and the poor prediction accuracy is reasonable. The GRU model considers the feature that traffic flow data are highly correlated in the temporal dimension and therefore has better prediction performance compared with DSC. In addition, since the DSC-GRU considers the spatiotemporal dependence of traffic flow data, the accuracy of the model is better than any of the components. In the first place, it proves the significance of considering the spatiotemporal dependence of traffic data to improve traffic flow prediction accuracy. In the second place, it demonstrates the effectiveness of the DSC-GRU model for solving the traffic flow prediction task.

7. Conclusions

In this paper, we propose a deep learning model, termed DSC-GRU, which can simultaneously consider the spatiotemporal characteristics of traffic flow data to deal with traffic flow prediction task. On the one hand, the DSC unit is used to capture the topological structure of the road network space and model the spatial dependence of traffic data. Unlike the traditional graph convolutional networks that only consider the characteristics of neighbor nodes, the DSC also considers the global characteristics of nodes in the whole road network space based on the correlation matrix. The gated mechanism is introduced to control the relative importance relationship between the adjacent information and the global information. The impact of the spatial dependence of the traffic road network are fully considered. On the other hand, the GRU is used to capture the characteristics of dynamic changes in traffic flow and to model the time dependence of traffic data. The DSC unit is embedded in the GRU network to add spatial information to each state in the GRU network. Based on two real-world datasets, parametric experiments and ablation experiments are conducted to select the optimal structure and parameters. Comparison experiments show that the DSC-GRU model has better prediction performance at different horizons and outperforms each of the comparison models. In conclusion, the DSC-GRU can be extended to other prediction tasks with spatiotemporal characteristics.

The DSC-GRU model still has some shortcomings. The influence of external factors is not considered in the model. The effects of weather, traffic accidents, and surrounding buildings should be considered to make the predicted traffic flow data more realistic. In subsequent work, the dataset containing the above information needs to be selected for further validation of our proposed model.

Author Contributions: Conceptualization, Qingyong Zhang and Lingfeng Zhou; data curation, Bingrong Xu; formal analysis, Yixin Su; funding acquisition, Yixin Su; investigation, Yixin Su; methodology, Qingyong Zhang; project administration, Yixin Su; resources, Bingrong Xu; software, Huiwen Xia; supervision, Qingyong Zhang; validation, Qingyong Zhang and Lingfeng Zhou; visualization, Qingyong Zhang and Huiwen Xia; writing—original draft, Lingfeng Zhou; writing—review and editing, Lingfeng Zhou. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant (62206204) and the Natural Science Foundation of Hubei Province grant number [2019CFB571].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: A publicly available dataset was analyzed in this study. It can be found here: https://github.com/wanhuaiyu/ASTGCN/tree/master/data, accessed on 1 June 2023.

Acknowledgments: The authors would like to thank the National Natural Science Foundation of China under Grant (62206204) and the Natural Science Foundation of Hubei Province grant number [2019CFB571] for supporting this research.

Conflicts of Interest: The authors declare no conflict of interest. The funder had no role in the design of the study, in the collection, analysis, or interpretation; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. The Pseudocode of DSC_GRU Training

Algorithm A1 The training of DSC_GRU

Input: Traffic flow data at t moment X_t. Initial hidden state H_1. Traffic flow true value Y. Traffic road network R_N.

Output: Predicted value Y_{out}. Training epochsN_{epoch}.

Initialize: The length of historical data used for prediction h. Hidden layer state $H_i (i \in (1, h])$. Process the adjacent graph $Adjacent(\cdot)$. Process the global graph $R^2(\cdot)$. Loss function $Loss(\cdot)$.

//Processing graph data
$G_a = Adjacent(R_N)$
$G_r = R^2(R_N)$
for epoch $e = 1$ to N_{epoch} **do**
  for timestep $i = 1$ to h **do**
    //Reset gate output
    $r_i \leftarrow \sigma(W_r \cdot [X_i, H_i] + b_r)$
    //Update gate output
    $u_i \leftarrow \sigma(W_u \cdot [X_i, H_i] + b_u)$
    //Graph Convolution
    $GCN_a \leftarrow GCN\big(G_a, X_i, (r_i * H_i)\big)$
    $GCN_r \leftarrow GCN\big(G_r, X_i, (r_i * H_i)\big)$
    $Fusion \leftarrow \sigma\big(W_o \cdot (GCN_a + GCN_r) + b_o\big)$
    $DSC \leftarrow \big(GCN_a * (1 - Fusion)\big) + (GCN_r * Fusion)$
    //Calculate the candidate hidden state
    $c_i \leftarrow tanh(W_c \cdot DSC + b_c)$
    //Calculate the candidate hidden state
    $H_{i+1} \leftarrow u_i * H_i + (1 - u_i) * c_i$
  end for
  $Y_{out} \leftarrow tanh(W_p H_{h+1} + b_p)$
  //Calculate the loss of DSC-GRU
  $L \leftarrow Loss(Y, Y_{out})$
end for
return $DSC - GRU$

References

1. Liu, Y.; James, J.; Kang, J.; Niyato, D.; Zhang, S. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet Things J.* **2020**, *7*, 7751–7763. [CrossRef]
2. Lin, X.; Sun, X.; Ho, P.H.; Shen, X. GSIS: A secure and privacy-preserving protocol for vehicular communications. *IEEE Trans. Veh. Technol.* **2007**, *56*, 3442–3456.
3. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
4. Kumar, S.V.; Vanajakshi, L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *Eur. Transp. Res. Rev.* **2015**, *7*, 21. [CrossRef]
5. Emami, A.; Sarvi, M.; Bagloee, S.A. Short-term traffic flow prediction based on faded memory Kalman Filter fusing data from connected vehicles and Bluetooth sensors. *Simul. Model. Pract. Theory* **2020**, *102*, 102025. [CrossRef]
6. Wang, H.; Liu, L.; Dong, S.; Qian, Z.; Wei, H. A novel work zone short-term vehicle-type specific traffic speed prediction model through the hybrid EMD–ARIMA framework. *Transp. B-Transp. Dyn.* **2016**, *4*, 159–186. [CrossRef]
7. Zhang, X.; He, G.; Lu, H. Short-term traffic flow forecasting based on K-nearest neighbors non-parametric regression. *J. Card. Surg.* **2009**, *24*, 178–183.
8. Petrlik, J.; Fucik, O.; Sekanina, L. Multiobjective selection of input sensors for svr applied to road traffic prediction. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Ljubljana, Slovenia, 13–17 September 2014; pp. 802–811.
9. Westgate, B.S.; Woodard, D.B.; Matteson, D.S.; Henderson, S.G. Travel time estimation for ambulances using Bayesian data augmentation. *Ann. Appl. Stat.* **2013**, pp. 1139–1161. [CrossRef]
10. Chen, X.; Wu, S.; Shi, C.; Huang, Y.; Yang, Y.; Ke, R.; Zhao, J. Sensing data supported traffic flow prediction via denoising schemes and ANN: A comparison. *IEEE Sens. J.* **2020**, *20*, 14317–14328. [CrossRef]

11. Cai, Q.; Abdel-Aty, M.; Sun, Y.; Lee, J.; Yuan, J. Applying a deep learning approach for transportation safety planning by using high-resolution transportation and land use data. *Transp. Res. Part A-Policy Pract.* **2019**, *127*, 71–85. [CrossRef]
12. Van Lint, J.; Hoogendoorn, S.P.; van Zuylen, H.J. Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks. *Transp. Res. Rec.* **2002**, *1811*, 30–39. [CrossRef]
13. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
14. Parizad, A.; Hatziadoniu, C. Deep Learning Algorithms and Parallel Distributed Computing Techniques for High-Resolution Load Forecasting Applying Hyperparameter Optimization. *IEEE Syst. J.* **2022**, *16*, 3758–3769. [CrossRef]
15. Zhang, Q.; Li, C.; Su, F.; Li, Y. Spatio-Temporal Residual Graph Attention Network for Traffic Flow Forecasting. *IEEE Internet Things J.* **2023**, *10*, 11518–11532. [CrossRef]
16. Liao, Z.; Huang, H.; Zhao, Y.; Liu, Y.; Zhang, G. Analysis and Forecast of Traffic Flow between Urban Functional Areas Based on Ride-Hailing Trajectories. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 144. [CrossRef]
17. Méndez, M.; Merayo, M.G.; Núñez, M. Long-term traffic flow forecasting using a hybrid CNN-BiLSTM model. *Eng. Appl. Artif. Intell.* **2023**, *121*, 106041. [CrossRef]
18. Peng, H.; Wang, H.; Du, B.; Bhuiyan, M.Z.A.; Ma, H.; Liu, J.; Wang, L.; Yang, Z.; Du, L.; Wang, S.; et al. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Inf. Sci.* **2020**, *521*, 277–290. [CrossRef]
19. Yin, X.; Wu, G.; Wei, J.; Shen, Y.; Qi, H.; Yin, B. Deep learning on traffic prediction: Methods, analysis, and future directions. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 4927–4943. [CrossRef]
20. Lin, G.; Lin, A.; Gu, D. Using support vector regression and K-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient. *Inf. Sci.* **2022**, *608*, 517–531. [CrossRef]
21. Zhang, Q.; Yin, C.; Chen, Y.; Su, F. IGCRRN: Improved Graph Convolution Res-Recurrent Network for spatio-temporal dependence capturing and traffic flow prediction. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105179. [CrossRef]
22. Nadarajan, J.; Sivanraj, R. Attention-Based Multiscale Spatiotemporal Network for Traffic Forecast with Fusion of External Factors. *ISPRS Int. J. Geo Inf.* **2022**, *11*, 619. [CrossRef]
23. Yue, W.; Zhou, D.; Wang, S.; Duan, P. Engineering Traffic Prediction with Online Data Imputation: A Graph-Theoretic Perspective. *IEEE Syst. J.* **2023**, 17, 4485-4496. [CrossRef]
24. Wang, T.; Zhang, B.; Wei, W.; Damaševičius, R.; Scherer, R. Traffic flow prediction based on BP neural network. In Proceedings of the International Conference on AIID, Guangzhou, China, 28–30 May 2021; pp. 15–19.
25. Huang, W.; Song, G.; Hong, H.; Xie, K. Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2191–2201. [CrossRef]
26. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.Y. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 865–873. [CrossRef]
27. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part C-Emerg. Technol.* **2015**, *54*, 187–197. [CrossRef]
28. Sun, P.; Boukerche, A.; Tao, Y. SSGRU: A novel hybrid stacked GRU-based traffic volume prediction approach in a road network. *Comput. Commun.* **2020**, *160*, 502–511. [CrossRef]
29. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
30. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27* .
31. Zheng, G.; Chai, W.K.; Katos, V.; Walton, M. A joint temporal-spatial ensemble model for short-term traffic prediction. *Neural Comput.* **2021**, *457*, 26–39. [CrossRef]
32. Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* **2017**, *17*, 818. [CrossRef]
33. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.c. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neural Inf. Process. Syst.* **2015**, *28.* . [CrossRef]
34. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [CrossRef]
35. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI Conference Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 922–929.
36. Huang, Y.; Weng, Y.; Yu, S.; Chen, X. Diffusion convolutional recurrent neural network with rank influence learning for traffic forecasting. In Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019 ; pp. 678–685.
37. Bao, Y.; Huang, J.; Shen, Q.; Cao, Y.; Ding, W.; Shi, Z.; Shi, Q. Spatial–Temporal Complex Graph Convolution Network for Traffic Flow Prediction. *Eng. Appl. Artif. Intell.* **2023**, *121*, 106044. [CrossRef]
38. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.
39. Song, C.; Lin, Y.; Guo, S.; Wan, H. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Hilton, NY, USA, 7–12 February 2020; Volume 34, pp. 914–921.

40. Hao, S.; Lee, D.H.; Zhao, D. Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system. *Transp. Res. Part C-Emerg. Technol.* **2019**, *107*, 287–300. [CrossRef]
41. Li, Z.; Han, Y.; Xu, Z.; Zhang, Z.; Sun, Z.; Chen, G. PMGCN: Progressive Multi-Graph Convolutional Network for Traffic Forecasting. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 241. [CrossRef]
42. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural Inf. Process. Syst.* **2016**, *29*. [CrossRef]
43. Engelmann, F.; Bokeloh, M.; Fathi, A.; Leibe, B.; Nießner, M. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In Proceedings of the IEEE Conference Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 9031–9040.
44. Gligorijević, V.; Renfrew, P.D.; Kosciolek, T.; Leman, J.K.; Berenberg, D.; Vatanen, T.; Chandler, C.; Taylor, B.C.; Fisk, I.M.; Vlamakis, H.; et al. Structure-based protein function prediction using graph convolutional networks. *Nat. Commun.* **2021**, *12*, 3168. [CrossRef] [PubMed]
45. Adem, K. Diagnosis of breast cancer with Stacked autoencoder and Subspace kNN. *Phys. A* **2020**, *551*, 124591. [CrossRef]
46. Chen, X.; Wang, X.; Yi, B.; He, Q.; Huang, M. Deep Learning-Based Traffic Prediction for Energy Efficiency Optimization in Software-Defined Networking. *IEEE Syst. J.* **2021**, *15*, 5583–5594. [CrossRef]
47. Guo, S.; Lin, Y.; Wan, H.; Li, X.; Cong, G. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Trans. Knowl. Data Eng.* **2021**, *34*, 5415–5428. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

Large-Scale Automatic Identification of Industrial Vacant Land

Yihao Sun [1], Han Hu [2], Yawen Han [1], Ziyan Wang [1] and Xiaodi Zheng [1,3,*]

[1] Department of Landscape Architecture, School of Architecture, Tsinghua University, Beijing 100084, China; sunyh20@mails.tsinghua.edu.cn (Y.S.); hanyw18@tsinghua.org.cn (Y.H.); ziyanw@mail.tsinghua.edu.cn (Z.W.)
[2] School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA; huhan.opt@gmail.com
[3] Key Laboratory of Eco Planning & Green Building (Tsinghua University), Ministry of Education, Beijing 100084, China
* Correspondence: xzheng@tsinghua.edu.cn

Abstract: Many cities worldwide have large amounts of industrial vacant land (IVL) due to development and transformation, posing a growing problem. However, the large-scale identification of IVL is hindered by obstacles such as high cost, high variability, and closed-source data. Moreover, it is difficult to distinguish industrial vacant land from operational industrial land based solely upon image features. To address these issues, we propose a method for the large-scale automatic identification of IVL. The framework uses deep learning to train remote-sensing images of potential industrial vacant land to generate a semantic segmentation model and further use population density and surface temperature data to filter model predictions. The feasibility of the proposed methodology was validated through a case study in Tangshan City, Hebei Province, China. The study indicates two major conclusions: (1) The proposed IVL identification framework can efficiently generate industrial vacant land mapping. (2) HRNet exhibits the highest accuracy and strongest robustness after training compared with other semantic segmentation backbone networks, ensuring high-quality performance and stability, as evidenced by a model accuracy of 97.84%. Based on the above advantages, the identification framework provides a reference method for various countries and regions to identify industrial vacant land on a large scale, which is of great significance for advancing the research and transformation of industrial vacant land worldwide.

Keywords: industrial vacant land; land use; satellite images; multi-source data; deep learning

1. Introduction

The effective transformation of urban vacant land (UVL) has broad prospects and significant advantages for improving urban economy, ecological environment, transformational development, and urban planning [1–5]. The term UVL is broad and diverse but is usually defined as under-utilized lands including bare soil, derelict land, abandon buildings and structures, brownfields, greenfields, uncultivated land or marginal agricultural land, and recently razed land [6]. With the advent of the post-industrial era, some cities have begun to restructure their economy and transform their industrial areas. As a result, industrial areas in many cities have closed down, creating large amounts of industrial vacant land (IVL) [7–9]. Some examples include the "rust-belt" and Youngstown in the USA, and the Ruhr in Germany [10,11]. Urban development and expansion have often revolved around industrial land, particularly in cities that sprang up as a result of industrial production. Areas of IVL are widely distributed in cities, particularly in city centers, occupying prime urban locations, triggering environmental degradation, economic decline, and social exclusion; thus, many cities are confronted with the pressing need for revitalization of their older town centers [12], posing a serious challenge to urban planning [7–9]. These points demonstrate that IVL, as a category of UVL, is a valuable land resource and its reuse can effectively undermine urban sprawl and promote sustainable

urban development [13–17]. Many IVL areas are regenerated into new sites that integrate new functions into the urban system, such as real estate that provides economic value and urban green spaces that provide ecosystem services for inhabitants. For example, in the 1960s, the decadent Ruhr industrial district in Germany underwent a comprehensive renovation in terms of industrial restructuring and the optimization of the ecological environment. By changing the function of the original buildings, facilities, and sites, the history of the industrial area was effectively recreated while providing green spaces for culture and recreational life. There are many opportunities to redevelop vacant land in terms of ecological and social value [1]. Identifying IVL is, therefore, a prerequisite for its reuse to improve urban spatial patterns. However, IVL identification is challenged by the low efficiency and accuracy of previous IVL identification methods and other hindrances related to limited land-use data from some areas that are not yet open source. Certain researchers assume that vacant land exhibits certain patterns, such as the presence of buildings, vacant parking lots, and sparse vegetation. They employ object-based classification techniques using various data sources, including Ikonos, QuickBird satellite imagery [18], and 3D laser scanning data [19]. However, these studies face challenges with low accuracy in distinguishing between bare soil/land, built-up areas, and vacant land [20]. These studies primarily rely on image classification alone to identify vacant land. In reality, the term "vacant land" encompasses specific information about land use that is influenced by human activity. Image classification alone cannot provide such information; it needs to be complemented with additional data sources [20]. Furthermore, commercial data is used due to its high spatial resolution, such as QuickBird images with a resolution of 0.6 m. Despite the possible detection of specific types of vacant land, the cost of the data is not negligible. Consequently, the development of an effective and rigorous identification method is therefore crucial to strategizing IVL reuse and urban planning.

There are currently two main methods for identifying UVL and IVL. One is the traditional manual identification method, which includes field research, questionnaires, visual identification using satellite images, and marking areas using cadastral data. This approach requires a large amount of manpower and time. As a consequence, the literature mostly contains small-scale studies of urban areas. Li et al. (2018) used high-resolution satellite images of Shanghai from 2000, 2005, and 2010 to identify the UVL distribution during each time period using manual visual interpretation [21]. Similar methods were used in case studies in Guangzhou [22], Changchun [23], and Atibaia [24]. Wang et al. [25] used remote-sensing (RS) imagery combined with field research and questionnaires to identify UVL in Wuhu. Song et al. [26] combined manual field surveys with open-source data (e.g., maps, Baidu heat maps, business data, and Baidu open street maps) to quantify the UVL of the Changchun City center. Despite the significance of this work, most previous studies have faced limitations. For example, they are (1) expensive in terms of labor and time, (2) provide uncertain identification results due to subjective differences in the identification criteria between auditors, and (3) result in research that is difficult to expand to regional and national scales owing to limited data availability and labor time constraints.

Two methods have gained widespread attention in recent years that offer substantial savings in labor and time costs while identifying UVL and IVL on a large scale and have a high measurement accuracy. These methods include (1) machine-learning-based artificial intelligence recognition and (2) big-data-based data fusion recognition. Xu and Ehlers [20] established a rule-based data fusion framework that integrates RS images, geographical information system (GIS) layers, and citizen science data using datasets such as Urban Atlas (UA), impermeable subsurface data, land cover (LC), government data, OpenStreetMap, Wiki data, and social media data. They proposed that different datasets should be chosen for different UVL types to perform data fusion and identification, and they applied the method to identify UVL in 63 regions in Germany. Mao et al. [27] developed a large-scale automatic UVL identification framework using semantic segmentation of high-resolution remote-sensing (HRRS) images and city stratification, applied the framework in a case

study of 36 major Chinese cities, and obtained accurate and efficient UVL identification results. These studies have important implications for the field.

With the rapid development of deep-learning techniques in computer vision, semantic segmentation is becoming increasingly used in the field of the built environment and RS image recognition, including FCL [28,29], UNet [30], SegNet [29,31], and DeepLab [32,33], which are potential tools for achieving large-scale and automatic UVL identification. Most studies have focused on automatic UVL identification based on satellite images. However, little work has been conducted on automatic IVL identification because the RS images of IVL present a variety of visual characteristics, and their "vacant" or "idle" status cannot be judged by satellite images alone. Inconsistent identification and determination criteria have also been used. Thus, it has so far been difficult to establish an accurate and efficient automatic IVL identification framework.

To confront these problems, we aim to put forth a comprehensive methodology and framework for identifying large-scale IVL utilizing open-source data. This proposed methodology can be readily adapted and implemented in different countries and regions, offering a novel approach for IVL identification and analysis. Specifically, we propose a method to identify IVL based on satellite RS images and multi-source data. The main goal of this study is to achieve the automatic identification of potential IVL based on semantic segmentation of Sentinel-2A RS images, and to determine the true IVL characteristics by removing still-operational industrial sites based on population density and surface temperature inversion data. We compared three semantic segmentation frameworks—HRNet, ResNet-UNet, and VGG16-UNet—and then built a potential IVL semantic segmentation model based on the best-performing HRNet and a filtering mechanism using human density and surface temperature. We validated the whole identification workflow in Tangshan City, Hebei Province, China. The experiments show that the identification model based on HRNet is accurate, stable, efficient, and robust, and the filtering mechanism can cover most of the research objectives. This identification framework provides a reference method for various countries and regions to identify IVL on a large scale, which is of great significance for advancing IVL and UVL research and practice worldwide.

2. Study Area and Data

Three resource-based cities in China were investigated to study a large-scale automatic detection framework of IVL (Figure 1). Resource-based cities represent a common urban typology within China's urban system and are characterized by a predominant reliance on the extraction and processing of nonrenewable resources (e.g., minerals, lumber, oil) within their respective regions. Such cities are often accompanied by a considerable amount of industrial infrastructure and industrial land. A total of 262 resource-based cities were identified in accordance with the National Sustainable Development Plan for Resource-Based Cities (2013–2020) published by the State Council in 2013, which were further categorized into four distinct types: growing (31), mature (141), declining (67), and regenerating (23). The industrial systems of resource-declining and resource-regenerating cities are often well-established and distributed in different provinces, with diverse core industrial types representing the industrial construction characteristics under different regional features and urban morphology backgrounds in China. For many resource-regenerating cities, some IVL may be reused and regenerated to reassume industrial or other functions such as shopping areas and industrial culture parks; however, they may retain specific industrial structures, making it difficult to distinguish operational industrial land from genuinely idle IVL simply by observing RS image characteristics, thereby increasing the complexity and challenges associated with automatic IVL detection.

We aim to make this automatic detection method broadly applicable and highly feasible. Therefore, we selected potential IVL from three declining cities (Xinyu, Wuhai, and Huangshi) and one regeneration city (Tangshan) as models for creating the semantic segmentation training dataset. The central urban area of Tangshan was selected as the verification case for the entire detection method. The four cities used for dataset creation are

located in three geographical regions in the west (Wuhai), middle (Xinyu and Huangshi), and east (Tangshan) of China. The diversity of surface features in the dataset was considered with respect to the various industrial systems established around iron mines, coal mines, and quarries, as well as the diversity of land-use characteristics in different cities. These characteristics are influenced by various factors such as different climatic conditions, urban culture, and planning and construction styles.

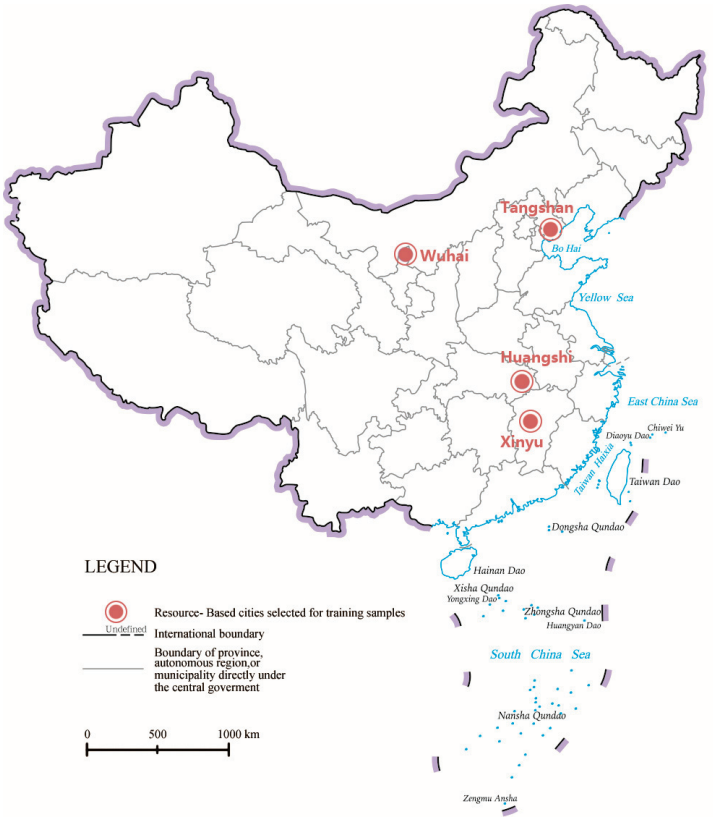


Figure 1. Locations of the four resource-based cities investigated here to study the large-scale automatic detection framework of IVL.

We collected Sentinel-2A RS images of the three cities from the European Space Agency (https://scihub.copernicus.eu/dhus/#/home, accessed on 11 November 2021). The images were synthesized from the red-green-blue channels to form false-color images with a pixel resolution of 10 m. Additionally, utilizing the bands 11 and 12 provided by the SWIR sensor aboard Sentinel-2A, we were able to ascertain the environmental temperature conditions at the target locations. This holds significant relevance in discerning the operational status of the land. These images are the highest-quality open-source and free RS data available. In contrast to lower-level L1C products that only undergo spatial correction, L2A-level images undergo atmospheric correction to obtain more accurate surface parameters. The data were collected from October 2020 to January 2021. The following criteria were followed for image selection to ensure the feasibility and data quality used in our method, including (1) low environmental temperature to ensure the exclusion of industrial sites that are still in operation using surface temperature data, and (2) cloud interference controlled at a very low level to allow clear analysis of land-use and land-cover features.

We also collected Landsat 8 RS images (https://earthexplorer.usgs.gov/, accessed on 23 December 2021) of the central urban area of Tangshan City, the validation target. The Band-10 data from the TIRS sensor were subjected to radiometric calibration. We computed the land surface temperature using the radiative transfer equation (RTE). Following this, we performed a merging process through the Environment for Visualizing Images (ENVI) platform's pansharpening technique. This involved integrating Band 2 (blue), Band 3

(green), and Band 4 (red) with a resolution of 30 m, along with Band 8 (pan) with a resolution of 15 m. This merging process provided a clear observation of the reasonableness of the inverted surface temperature raster data. Simultaneously, we obtained the Baidu heat map of the validation target during working hours from Baidu Map (https://map.baidu.com/, accessed on 12 April 2022) to calculate the population density in the city. The Baidu heat map can be converted into vector points representing specific population densities through vector calculations. The main calculation process involves using vectorization operations in ArcGIS to estimate the population distribution proportions provided by the Baidu Heat Map. This is combined with the total urban population to perform proportional sampling, thereby deriving the population density.

3. Methods

The entire automatic detection method includes two main stages: (1) predictive recognition of potential IVL through semantic segmentation models, followed by (2) IVL filtering using population density data and land surface temperature data. To ensure the accuracy of the semantic segmentation model and the feasibility of the subsequent filtering process, we developed a four-step framework (Figure 2) as follows: (1) data labeling, (2) model training, (3) potential IVL prediction, and (4) multi-source data filtering. The goal and contents of each step are described below.

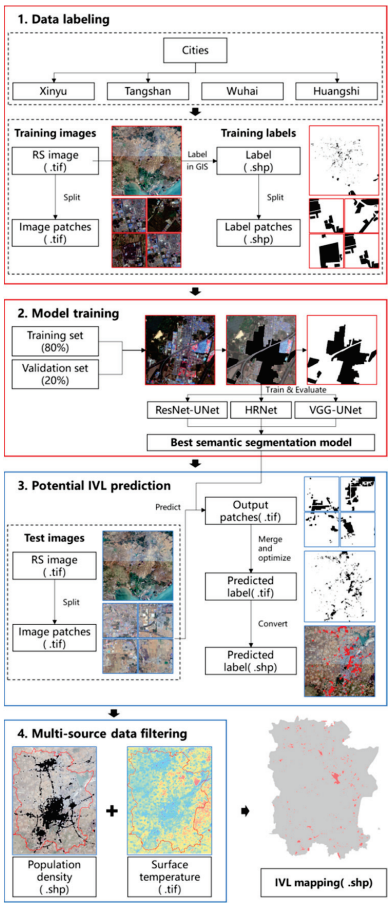


Figure 2. Large-scale automatic detection framework of IVL. The red boxes refer to model establishment and the blue boxes refer to prediction and filtering.

3.1. Data Labeling

The purpose of this step is to obtain high-quality training data for the semantic segmentation model of potential IVL. Labels were created for potential IVL with industrial surface features in the three investigated cities. To ensure effective training results, these

labels were required to meet the following three conditions: (1) cover a large number of potential IVL with different surface features, (2) ensure a wide range of industrial and non-industrial surface changes to enhance the robustness of the trained model, and (3) ensure a sufficient number of samples to control overfitting.

The dataset for potential IVL was accordingly created from RS images and potential IVL labeled by professional auditors. It should be noted that the surface features of IVL may vary owing to different industrial production tasks before the sites were abandoned (Table 1). The potential IVL in the four representative cities exhibits a wide range of these features, ensuring a selection principle of multiple types of training samples within the same category.

Table 1. Common and different aspects of the three categories of potential IVL: (a), (b), and (c) are the common surface features of the three categories; and (d) and (e) are the unique features presented by each category. We employed high-resolution imagery for effective visualization, and the images within the table were sourced from Google Earth Pro.

Category	Surface Features	Representative Remote-Sensing Image
All in common	(a) Boundaries are clear and generally demarcated by walls and roads. (b) Flat terrain. (c) Internal transportation system connects the entire venue.	
Raw material processing and manufacturing sites	(d) Features buildings with distinct industrial characteristics, such as blast furnaces, chimneys, and pipelines at steel plants. (e) May have identifying colors, such as grayish-white for cement plants, brownish for steel plants, and jet-black for coking plants.	
General manufacturing sites	(d) Has raw material and miscellaneous item storage but does not have large smelting. (e) Has large areas containing sheds with blue or red steel roofs or solar panels.	
Infrastructure and warehouse	(d) May contain railway tracks, numerous containers, storage tanks, and a barge wharf on a riverbank.	

After labeling the potential IVL of the three cities, we used label vector data as the reference extent. Subsequently, we employed the Split Raster tool within the GIS platform to segment the IVL label vector data and RS images in ArcGIS into 256 × 256 pixel rasters. Within each label raster, grid cells featuring industrial land surfaces were designated with a value of 1, while urban background grid cells received a value of 0. This approach not only ensured the effective utilization of data by accommodating partial overlaps at edges but also maintained a coherent correspondence between the label raster and RS image remote-sensing imagery. A total of 1246 patches were generated. The segmented dataset was randomly divided into a training set and a testing set with an 8:2 ratio, for the subsequent model training.

3.2. Model Training

Let us suppose that $L = \{(x_i, y_i)\}$ is a labeled training image, where x_i is the ith pixel in that image. Each sample pixel, $x_i \in \chi$, in the image is assigned with a ground truth label $y_i \in \{0, 1\}$, where $y_i = 0$ indicates that the pixel is the background and $y_i = 1$ indicates that the pixel belongs to the industrial surface class. The goal of our semantic segmentation task is to obtain a segmentation model $f_\theta : \chi \longrightarrow \{0, 1\}$ from the labeled training dataset, where $\theta \in \Theta$ is the learned model parameter.

Many existing segmentation networks, such as UNet, were used to convert the entire image into a low-resolution visual representation before upsampling it back to the original high-resolution size to achieve pixel-level semantic segmentation. However, some positional information may become mismatched during the upsampling process in these networks. Our semantic segmentation task is highly sensitive to position. To address the issue of inaccurate segmentation in these networks, we used the HRNet [34] network, which preserves high-resolution visual representations during the forward propagation process, thus effectively retaining positional information.

Figure 3 provides an overview of the HRNet model, which starts with a high-resolution convolutional stream and progressively incorporates additional convolutional streams of decreasing resolution, connecting these multi-resolution streams in parallel. The segmentation result is generated by concatenating the representations from multi-resolution convolutional streams.

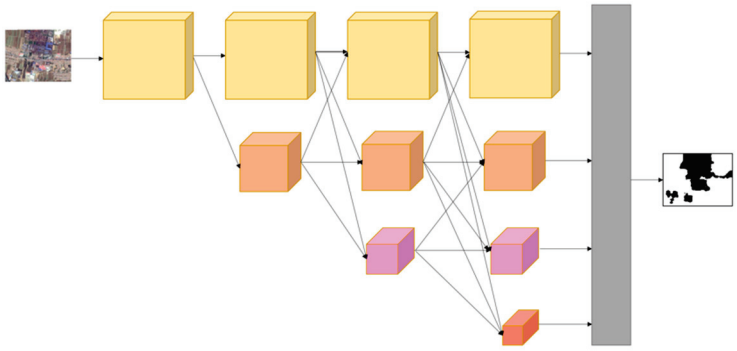


Figure 3. Overview of the HRNet model.

This HRNet model was trained from scratch without using any pretrained models. The loss function used to train the HRNet is binary cross entropy loss, which is commonly used in classification tasks. We used the Adam algorithm with an initial learning rate of 10^{-2}. The training epoch was set to 50.

3.3. Potential IVL Prediction

The purpose of this step is to obtain high-quality potential IVL prediction results in the form of vector data. We used the following methods to accomplish this goal and ensure the quality of the final results: (1) During model training, we installed the GDAL library for Python, enabling the use of TIFF data for both image input during model training

and the segmentation results of the model output. (2) We found fine points on both the edges and interior of the images. To reduce the amount of potential error when converting raster to vector files, we employed the ArcScan tool for raster refinement procedures. A morphological closing operation was performed on the predicted IVL raster. This operation effectively smooths out coarse boundaries within the raster while also appropriately filling small gaps between foreground objects, aiming to yield a raster with distinct boundaries and comprehensive internal filling. This practice is also advantageous for subsequent conversion into vector files, contributing to the creation of more precise potential IVL polygon vectors (Figure 4). Regarding the transformed outcomes, it was observed that, despite the morphology operations, certain independent and significantly small portions persisted within the vectorized polygons. Upon meticulous observation, statistical analysis, and considering the subsequent research's data requisites, independent polygons with an area of less than 400 m^2 were identified as noise and subsequently removed.



Figure 4. Comparison between optimized and non-optimized results: (**a**) original image; (**b**) ground truth; (**c**) non-optimized result with spots (red circles) on the edges; (**d**) optimized result with clear edges.

The process described of this section is shown in Figure 2: (1) Train a semantic segmentation dataset with the best-performing model architecture to obtain the prediction model and output the potential IVL prediction result raster tail. (2) Combine these tails into a complete raster. (3) Use image morphology operations, such as opening and closing in ArcScan, to optimize the raster. (4) Convert TIFF to SHP files. (5) Delete small-area polygons deemed as noise after area calculation.

3.4. Extraction of IVL

The potential IVL results obtained in the previous stage through semantic segmentation prediction consist of two parts: (1) industrial land that is still in use and (2) IVL. By analyzing the usage characteristics of these two types of land, we found that IVL is usually unoccupied or has very few people present during working hours. Additionally, the thermal radiation exhibited by buildings or structures within IVL sites is also often significantly lower than that of industrial land that is still in use, because heat is frequently generated during industrial processes. This phenomenon is more pronounced in northern China in winter owing to heating requirements in the cold climate. Two significant IVL characteristics are therefore sparse population density and low surface temperature. Considering that industrial production may require only a small number of on-site personnel, owing to a high degree of automation, the IVL identification criteria were set as follows. The population density of IVL is significantly lower than surrounding areas, and the maximum surface temperature within the land plot is also significantly lower than the average surface temperature of industrial areas that are currently operational.

At this stage, we utilized population density data and surface temperature data as the foundational elements for our analysis. Population density data were used for the initial filtering, followed by surface temperature data for further filtering. This systematic approach allowed us to separate operational industrial land from potential IVL, thus obtaining the recognition results for IVL. The filtering logic is shown in Figure 5.

The central urban area of Tangshan is taken as an example in this study to determine the target recognition area. The data used in this case study were from a statutory workday

(Friday, 18 December 2020). This specific day was chosen to ensure that the population distribution data were not affected by weekend or holiday features, which could lead to an underestimation of the true population distribution on active industrial land, thereby resulting in their misclassification as IVL. Conducting this study in winter also ensures that operating industrial sites can be distinguished from IVL based on their significantly higher surface temperatures. This distinction arises because heat radiation is not emitted from IVL owing to the absence of on-site production and heating.

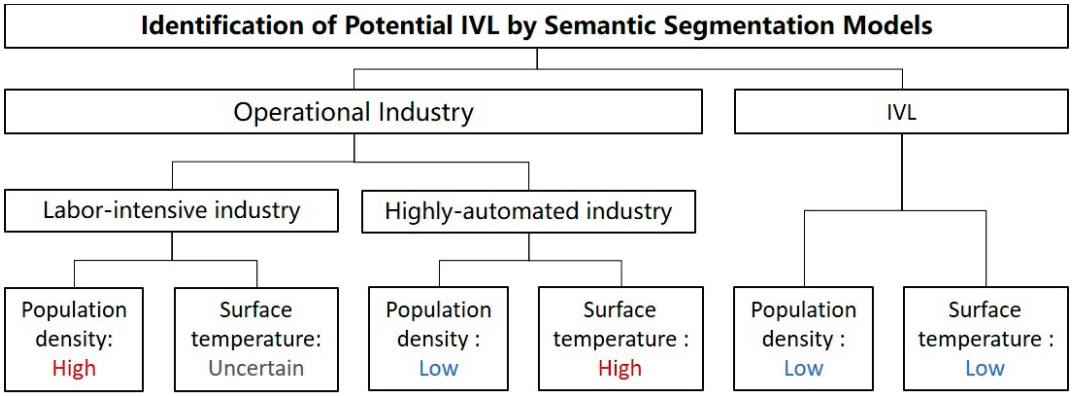


Figure 5. Logic of multi-source data filtering.

Due to the restricted accuracy of RS images, the semantic segmentation prediction results struggle to accurately separate large potential IVL in accordance with the city road configuration. Factory or enterprise areas are rarely separated by urban expressways or highways. To prevent scenarios where a large piece of land is mistakenly identified as IVL due to an abnormally high population density or surface temperature during the subsequent filtering process, this section uses vector data of Tangshan's major transportation network (including railways, highways, expressways, and main roads). This transportation network data are employed to separate the area identified as potential IVL (Figure 6).



Figure 6. Potential IVL after road division: (**a**) original image; (**b**) prediction results before division; (**c**) optimized result based on a typical road network layout, the extensive potential IVL has been divided into 7 independent parcels.

The population density data obtained from vectorizing the Baidu heat map were imported into GIS (Figure 7). The vectorization process of Baidu heat maps involves the following procedural steps, including (1) reclassifying the thermal raster data acquired from the Baidu Maps web service to extract thermal values at varying levels; (2) converting the thermal raster image into pixel points and estimating population data by deducing RGB values corresponding to the point positions in the heat map; and (3) subdividing the research area into 100 × 100 m grid cells, aggregating pixel-based population data within their respective grid cells, and further expanding the vectorized grid-based population using the urban population figures from the Seventh National Census Data provided by

the National Bureau of Statistics (http://www.stats.gov.cn/sj/pcsj/rkpc/d7c/, accessed on 25 May 2023), thus yielding population density data measured in people per hectare (people/hm^2). After vectorization, each vector point in the layer possessed a POP value representing a population density within a one-hectare square centered on the vector point. If the population density was less than 100 people/hm^2, the POP value was recorded as zero and excluded from the vectorization calculation results. Statistical analysis showed that the maximum population density within the central urban area of Tangshan at that time was 2277 people/hm^2, the minimum was 247 people/hm^2, and the average was 888.01 people/hm^2. In China, some control indicators of industrial land (e.g., plot ratio, building coefficient) must comply with the Industrial Project Construction Land Control Indicators (Ministry of Natural Resources of the People's Republic of China 2021, http://gk.mnr.gov.cn/zc/zxgfxwj/202102/t20210226_2615419.html, accessed on 25 May 2023). Under these requirements, the population density of labor-intensive industries is usually set to 120–140 people/hm^2. We therefore assumed a conservative scenario where the population density within IVL was less than 120 people/hm^2, and used this as a criterion to filter out a set of IVL and highly-automated industry land.

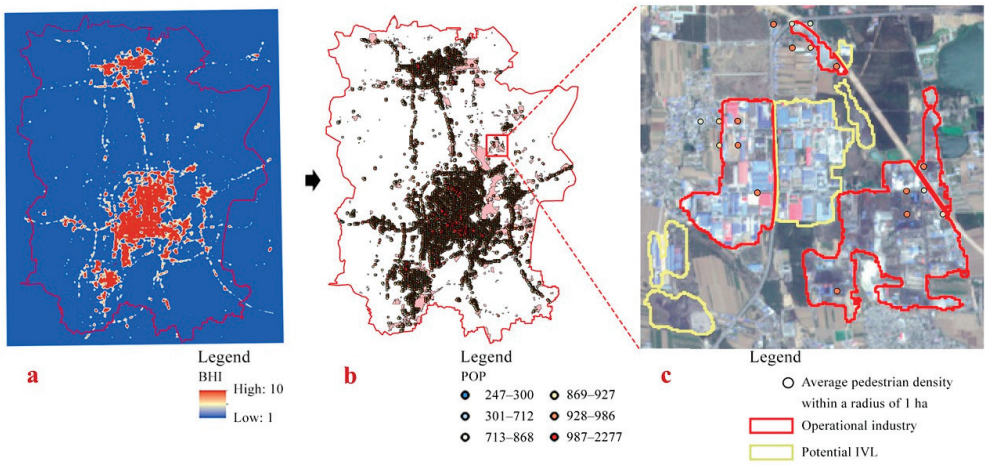


Figure 7. Filtering is first applied using population density data. (**a**) Performance on the client side of the Baidu heat map. (**b**) Population density vector point data after vectorization calculation. (**c**) Difference in population density between operational industry and potential IVL.

We utilized the spectral data from Landsat 8 images as the foundational basis, employing the radiative transfer equation (RTE) to perform the retrieval of regional land surface temperature [35]. This methodology primarily relies on the observation of surface thermal radiation Q by the satellite's infrared sensor (data encompassed by Band 10). Subsequently, atmospheric influences are subtracted, resulting in the determination of the intensity of surface thermal radiation. Finally, this intensity of thermal radiation undergoes conversion to ascertain the land surface temperature. The formulation of this method is presented as:

$$Q = [xD(t) + (1 - x)q_1]\alpha + q_2 \quad (1)$$

$$t = k_2 / \ln(k_1 / D(t) + 1) \quad (2)$$

In the given formulas, x represents the land surface emissivity, t denotes the land surface temperature, $D(t)$ signifies the blackbody radiance, and α signifies the atmospheric transmissivity in the thermal infrared domain. q_1 represents the atmospheric upwelling radiance, while q_2 denotes the atmospheric downwelling radiance. For distinct Landsat data, the values of k_1 and k_2 vary. For Band 10 of TIRS, k_1 is equal to 774.89 W·m^{-2}·sr^{-1}·μm^{-1}, and k_2 equals 1321.08 K.

Following the aforementioned methodology, we obtain the surface temperature of the study area, which ranged from −7.3 °C to 19.2 °C. Statistical analysis showed that

the lowest surface temperature of operational industrial land with a human population density higher than 120 persons per hectare was 8 °C. Considering the winter research context, we inferred that sites with a surface temperature above 8 °C were indicative of thermal radiation generated by ongoing industrial production. We were consequently able to identify highly-automated industrial land that has a low human population density but is still engaged in industrial production (Figure 8). The contours of the buildings depicted in the figure were sourced from BIGEMAP (http://www.bigemap.com/, accessed on 28 May 2022), whereby the vector tools for buildings facilitate the provision of real-time vector data derived from the data source of Gaode Map (https://www.amap.com/, accessed on 28 May 2022). This serves as a means to examine and interpret extensive temperature anomalies, thereby aiding in the assessment of the logicality of such deviations, and additionally contributing to the assessment of the operational status of the sites where the buildings are situated.

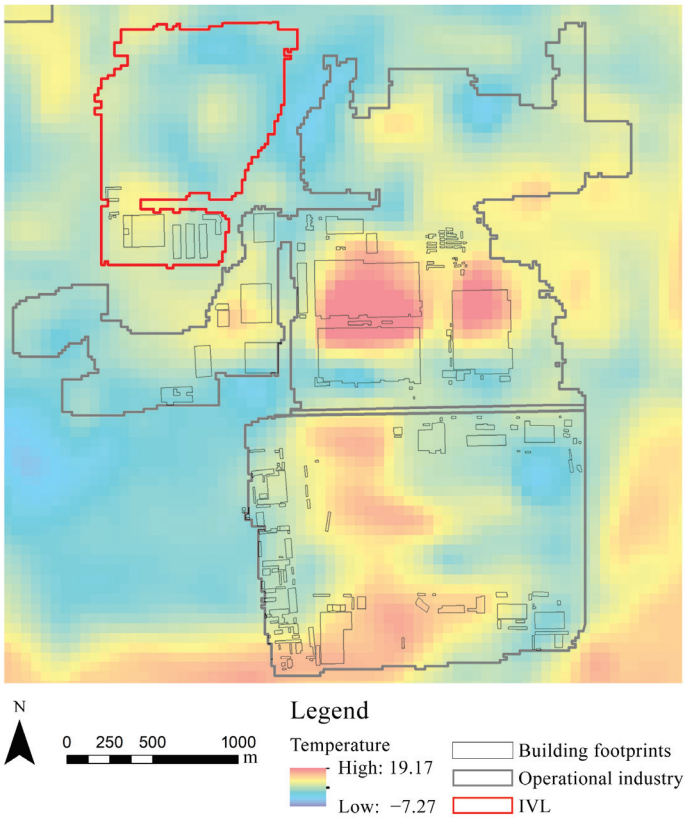


Figure 8. Difference in land surface temperature between land with operational industries and IVL.

4. Results

4.1. Overview of Extracted Sites

After completing the entire workflow, we mapped the industrial land still in operation and IVL in the central urban area of Tangshan City, as shown in Figure 9. The number, area, and geographic distribution of IVL, as well as the vacancy rate of urban industrial land were obtained from the results as expected. We identified 592 potential IVL areas in the central urban area of Tangshan. After screening with surface temperature and population distribution data, 431 IVL areas were extracted, covering an area of approximately 24.2 km^2. The results show that approximately 34% of the total industrial land in the central urban area of Tangshan is vacant.

Figure 10 shows a boxplot analysis of the area of the two land types. Although the number of IVL fields is significantly greater than that of operational industries, the maximum, minimum, upper quartile, lower quartile, and median areas of IVL are significantly lower (Figure 10). It can thus be concluded that as urban industrial transformation ad-

vances, small factories in Tangshan City are more likely to shut down than large factories, resulting in a large amount of small IVL scattered throughout the city. The reasons for this phenomenon may include a single production business, lack of capital and assets leading to difficulties in capital turnover, poor product quality, and/or inadequate employee welfare [36,37].

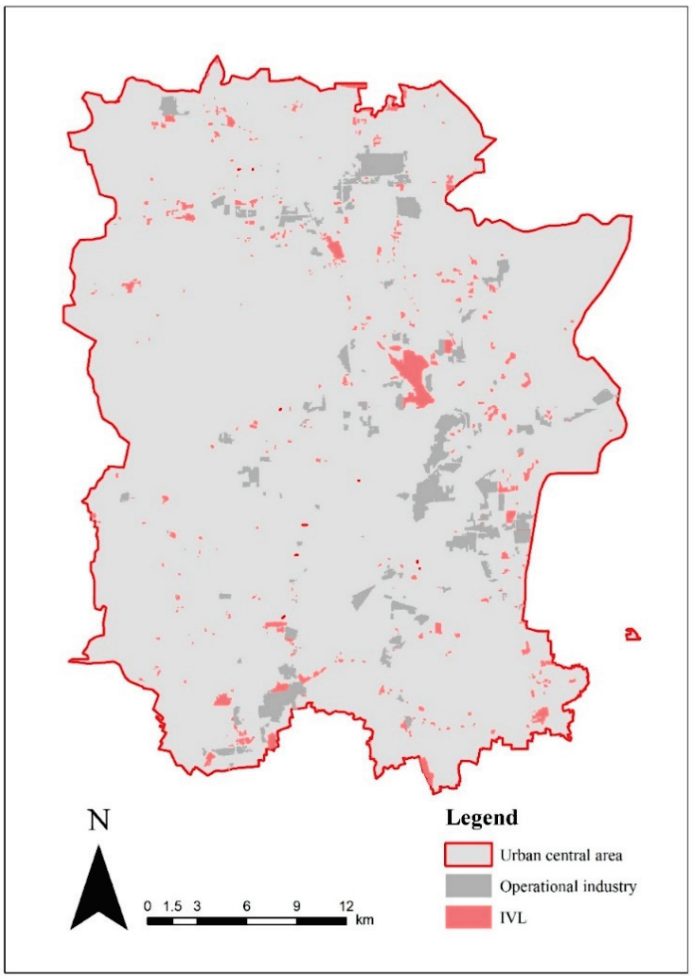


Figure 9. Mapping of IVL in the central urban area of Tangshan.

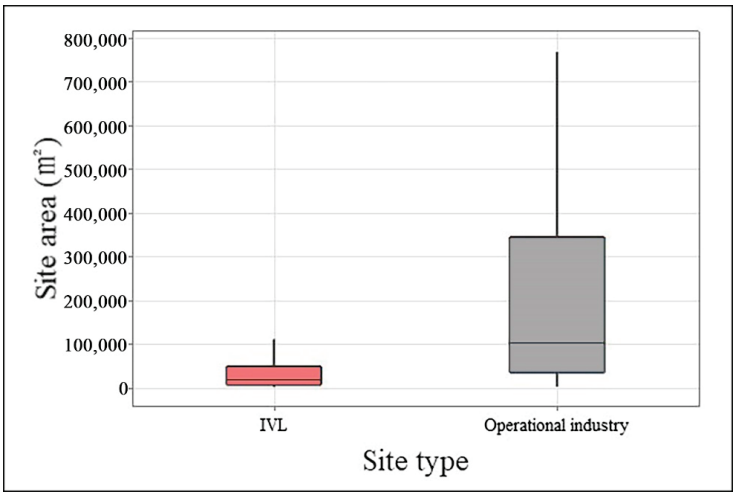


Figure 10. Site area differences between IVL and operational industries.

4.2. Model Evaluation

The ground-truth label for the test data is provided; thus, we employed three measures to evaluate the performance of the model: precision, recall, and F1-scores. The calculation formulas of the three metrics are as follows.

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1 - score = \frac{2\ precision * recall}{precision + recall} \tag{5}$$

where *TP* and *FN* represent the number of pixels belonging to the industrial surface that are correctly and incorrectly classified, respectively, and *FP* represents the number of pixels that belong to the background and are incorrectly classified. *Precision* represents the percentage of the correctly classified number of pixels in the predicted results, *recall* represents the percentage of the number of pixels classified in the ground truth, and *F1-score* refers to a tradeoff measure between the precision and recall. Therefore, one goal of a good classifier is to maximize the F1-score to a value of one. The HRNet evaluation results from the testing data are listed in Table 2.

Table 2. Performance evaluation of HRNet.

	Precision	**Recall**	**F1-Score**
HRNet	0.7782	0.7501	0.7640

4.3. Comparison Study

4.3.1. Comparison with Other Models

We compared the performance of HRNet to the model structure from two previous studies to examine the effect of the model structure. These methods include ResNet-UNet [38] and VGG16-UNet [39], which are commonly applied for semantic segmentation. The ResNet-UNet architecture uses the first four layers of ResNet50 for downsampling and replaces the transposed convolution using Pixel Shuffle in the upsampling. The VGG16-UNet architecture uses the UNet structure and uses the VGG-16 network as the encoder. The main difference between these two networks and our HRNet is that the previous methods first downsample the input image as a low-resolution representation, whereas our model maintains high-resolution representation throughout the entire forward process.

For a fair comparison, we trained each model from scratch for 50 epochs without pretraining. The initial learning rate was set to 10^{-2} and the Adam algorithm was used. The results are shown in Table 3, demonstrating that HRNet significantly outperformed the others in F1-score.

Table 3. Comparison of the three evaluation metrics in the three models.

	Precision	**Recall**	**F1-Score**
HRNet	0.7782	0.7501	0.7640
ResNet-UNet	1.0	0.4829	0.6513
VGG16-UNet	1.0	0.4829	0.6513

Figure 11 demonstrates that the accuracy of HRNet continues to increase and reaches nearly 1 at the end of training, whereas the accuracies of ResNet-UNet and VGG16-UNet increase over the first few epochs but then remain essentially constant at an accuracy of approximately 0.93. It is notable that the value of accuracy is much higher than precision and recall. This may be because the background pixels dominate the dataset, making it

easier for the model to learn the features of the background and predict most pixels as background, resulting in a higher accuracy rate.

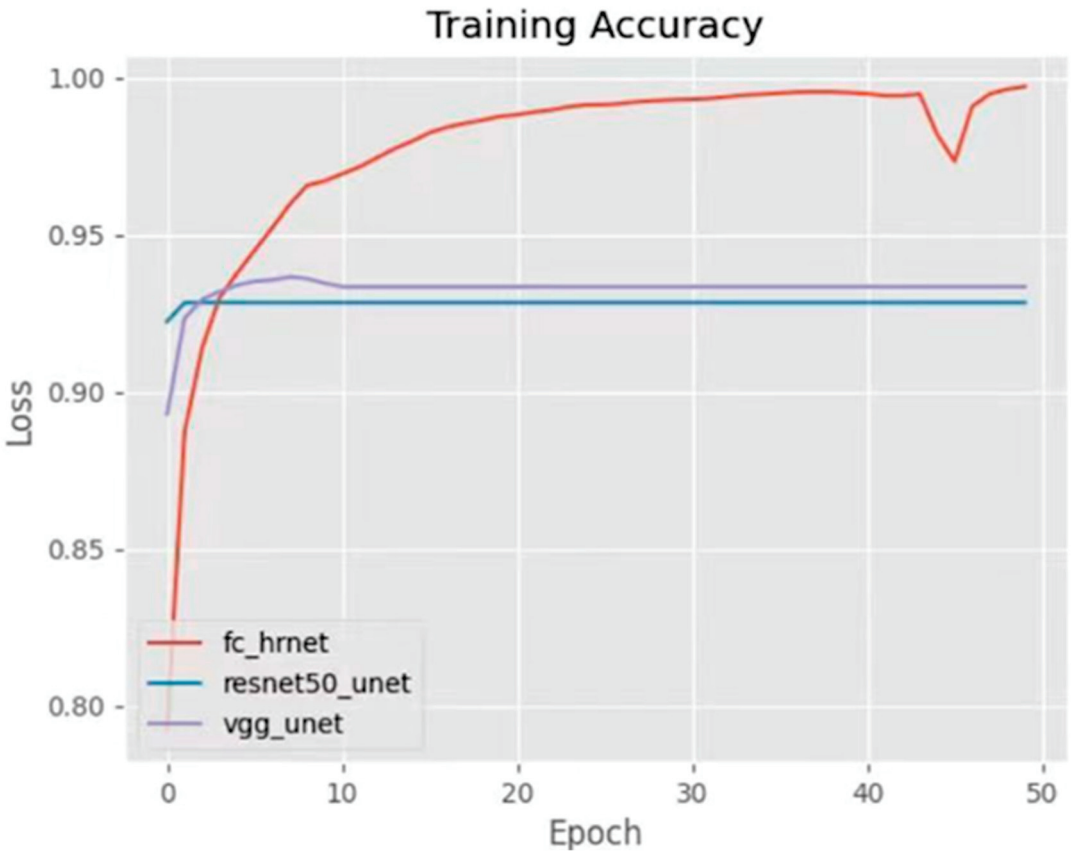


Figure 11. Comparison of accuracy curve of the three models.

Table 3 compares the performance of the three trained segmentation networks using the same training and testing datasets. When using ResNet-UNet and VGG16-UNet, a precision value of one and a low recall value indicate that these models classify almost all pixels as positive during segmentation, which suggests that these models are overfitting to the negative class. The F1-score provides a balanced evaluation of both precision and recall. HRNet shows the highest F1-score among all three models, which indicates that our proposed HRNet has an exceptional capacity to segment both positive and negative classes in a general sense.

4.3.2. Comparison with Traditional Identification Methods

Compared with traditional IVL identification methods, the innovation of this large-scale automatic detection framework lies primarily in the advantages demonstrated by the potential IVL semantic segmentation model. These include (1) the input data are highly accessible and can be used for more cities, covering a larger spatial range, (2) the method improves efficiency and reduces manpower, and (3) the conclusion demonstrates a significantly elevated accuracy comparable to manual recognition, while also maintaining a commendable level of stability and reliability and being unaffected by subjectivity owing to different auditors.

Traditional identification methods are heavily reliant on visual identification and resort to higher-resolution but non-open or commercially obtainable remote-sensing data, often entailing costs for annotators' convenience [27]. Moreover, outcomes from such visually driven conventional approaches typically pertain exclusively to specific regions for UVL identification. Consequently, traditional recognition methods impose stringent data prerequisites and exhibit limitations in terms of usage scenarios. Our innovation lies in

leveraging the superior performance of HRNet within the domain of image segmentation. By utilizing freely accessible and open-source Sentinel-2A data, we have achieved noteworthy image segmentation outcomes. Additionally, our approach demonstrates novelty through considerations of geographical diversity during the dataset construction phase, thereby rendering the model versatile across a wider array of usage scenarios.

In terms of efficiency comparison, we conducted an analysis by contrasting the total time required for the manual annotation of potential IVL across four cities with the combined time for model training and prediction. The outcomes demonstrate that the collective time for manually annotating potential IVL in these four cities amounts to approximately 50 h. Conversely, when leveraging the computational capabilities of one Nvidia RTX 3090 GPU within the automated framework for both training and prediction, the time required is approximately 4.5 h, representing an 11-fold enhancement in efficiency. Notably, if the computational power of the GPU employed for model training is augmented or if a multi-GPU configuration is employed, the efficiency of the automated framework would experience significant improvement. Moreover, when confronted with larger recognition areas, the efficiency advantages of the automated framework will become even more conspicuous.

We employed the intersection over union (IoU) value, which is the most widely adopted evaluation metric for the similarity between two image segmentation results, to contrast the differences in accuracy between the automated framework and traditional methods [40,41]. The equation for IoU value is as follows:

$$IoU(A, B) = area(A \cap B)/area(A \cup B) \tag{6}$$

During the dataset creation phase, the identification of potential IVL in the four cities was carried out by four professional auditors. In this phase of accuracy comparison and validation, we rotated the cities assigned to each of the four auditors. Subsequently, another round of visual identification for IVL was conducted across the four cities. The IoU value was calculated as a metric for benchmarking the accuracy of manual annotation, resulting in a value of 73.28%. Contrasting this outcome with the IoU achieved by the HRNet model after training, the IoU achieved by the automated framework reached 63.69%, achieving the human-derived metric by 86.91%. This substantiates that the automated recognition accuracy aligns with the proficiency of professional annotators engaged in visual identification. Simultaneously, the occurrence of variations in manual IoU values underscores the level of susceptibility to subjective influence from annotators.

5. Conclusions and Outlooks

In this study, we propose a framework for the large-scale automatic detection of IVL based on semantic segmentation of RS images and multi-source data filtering, including urban surface temperature and urban population density data. We selected the HRNet backbone network with the best performance after comparative experiments to train a semantic segmentation model, and verified its reliability and stability. We then validated the workflow using the central urban area of Tangshan City as an example and explained the distribution characteristics of IVL in the region.

In addition to the innovation of the semantic segmentation model, the introduction of multiple data sources ensures that the large-scale automatic IVL detection results are more reasonable and accurate. The segmentation of roads helps to extract IVL in a way that conforms to the basic logic of urban land use, which compensates for the resolution shortcomings of open and free RS images. The introduction of temperature and population density data enables operating industrial sites and IVL to be more easily distinguished. We must make clear, however, that a small number of special industrial types and industrial buildings with good insulation may result in an anomalously lower surface temperature, which could cause a small amount of error in the filtering of operating industrial land. However, for a large number of research targets, the overall framework can accurately map IVL in any city or region with sufficient open-source data.

Several problems remain to be solved to improve the model training results and IVL extraction, including (1) incorporating more bands of RS images to enhance the feature extraction of potential IVL; (2) generalizing the method to high-resolution RS imagery and more precise population density data for specific needs; and (3) increasing the sample size to include more cities to further enrich the training set and improve the model accuracy.

Author Contributions: Conceptualization, Yihao Sun and Xiaodi Zheng; data curation, Ziyan Wang and Yawen Han; formal analysis, Yihao Sun; funding acquisition, Xiaodi Zheng; methodology, Yihao Sun and Han Hu; project administration, Yihao Sun; software, Yihao Sun and Han Hu; supervision, Xiaodi Zheng; validation, Yihao Sun; visualization, Yihao Sun; writing—original draft, Yihao Sun and Yawen Han; writing—review and editing, Yihao Sun, Yawen Han, and Xiaodi Zheng. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Key Research and Development Program of China (No.2020YFC1807505).

Data Availability Statement: Supplementary data to this article are available online at https://data.mendeley.com/datasets/jzvcgcxd9f/2 for details (accessed on 1 September 2023).

Acknowledgments: The authors would like to thank National Key Research and Development Program of China for helpful supports related to this work. The authors would like to thank the anonymous reviewers for their helpful remarks.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kim, M.; Kim, G. Typology of urban left-Over space. In Proceedings of the Conference abstracts CELA Annual Meeting, Champaign, IL, USA, 28–31 March 2012.
2. Kremer, P.; Hamstead, Z.A.; McPhearson, T. A social-ecological assessment of vacant lots in New York City. *Landsc. Urban Plan.* **2013**, *120*, 218–233. [CrossRef]
3. Ardiwijaya, V.S.; Sumardi, T.P.; Suganda, E.; Temenggung, Y.A. Rejuvenating idle land to sustainable urban form: Case study of Bandung metropolitan area, Indonesia. *Procedia Environ. Sci.* **2015**, *28*, 176–184. [CrossRef]
4. Martinat, S.; Dvorak, P.; Frantal, B.; Klusacek, P.; Kunc, J.; Navratil, J.; Osman, R.; Tureckova, K.; Reed, M. Sustainable urban development in a city affected by heavy industry and mining? Case study of brownfields in Karvina, Czech Republic. *J. Clean. Prod.* **2016**, *118*, 78–87. [CrossRef]
5. Smith, J.P.; Li, X.; Turner, B.L., II. Lots for greening: Identification of metropolitan vacant land and its potential use for cooling and agriculture in Phoenix, AZ, USA. *Appl. Geogr.* **2017**, *85*, 139–151. [CrossRef]
6. Pagano, M.A.; Bowman, A.O.M. *Vacant Land in Cities: An Urban Resource*; Brookings Institution, Center on Urban and Metropolitan Policy: Washington, DC, USA, 2000.
7. Liebmann, H.; Kuder, T. Pathways and Strategies of Urban Regeneration-Deindustrialized Cities in Eastern Germany. *Eur. Plan. Stud.* **2012**, *20*, 1155–1172. [CrossRef]
8. Nassauer, J.I.; Raskin, J. Urban vacancy and land use legacies: A frontier for urban ecological research, design, and planning. *Landsc. Urban Plan.* **2014**, *125*, 245–253. [CrossRef]
9. Rieniets, T. Shrinking Cities: Causes and Effects of Urban Population Losses in the Twentieth Century. *Nat. Cult.* **2009**, *4*, 231–254. [CrossRef]
10. Reckien, D.; Martinez-Fernandez, C. Why Do Cities Shrink? *Eur. Plan. Stud.* **2011**, *19*, 1375–1397. [CrossRef]
11. Wiechmann, T.; Pallagst, K.M. Urban shrinkage in Germany and the USA: A Comparison of Transformation Patterns and Local Strategies. *Int. J. Urban Reg. Res.* **2012**, *36*, 261–280. [CrossRef]
12. Sołtysik, M.; Mazur-Belzyt, K. City Space Recycling: The Example of Brownfield Redevelopment. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *960*, 042016. [CrossRef]
13. Lange Deborah, A.; McNeil, S. Brownfield Development: Tools for Stewardship. *J. Urban Plan. Dev.* **2004**, *130*, 109–116. [CrossRef]
14. Nagengast, A.; Hendrickson, C.; Lange, D. Commuting from US Brownfield and Greenfield Residential Development Neighborhoods. *J. Urban Plan. Dev. Asce* **2011**, *137*, 298–304. [CrossRef]
15. De Sousa, C.A. Unearthing the benefits of brownfield to green space projects: An examination of project use and quality of life impacts. *Local Environ.* **2006**, *11*, 577–600. [CrossRef]
16. Baing, A.S. Containing Urban Sprawl? Comparing Brownfield Reuse Policies in England and Germany. *Int. Plan. Stud.* **2010**, *15*, 25–35. [CrossRef]
17. Potočnik, J. Addressing soil contamination, land take and soil sealing: Good for the environment, good for the economy. In Proceedings of the Conference 'Soil Remediation and Soil Sealing', Brussels, Belgium, 10 May 2012.

18. Banzhaf, E.; Netzband, M. Detecting urban brownfields by means of high resolution satellite imagery. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2004**, *35*, 460–466.
19. Li, X. Use of LiDAR in Object-based Classification to Characterize Brownfields for Green Space Conversion in Toledo. Ph.D. Thesis, University of Toledo, Toledo, OH, USA, 2017.
20. Xu, S.J.; Ehlers, M. Automatic detection of urban vacant land: An open-source approach for sustainable cities. *Comput. Environ. Urban Syst.* **2022**, *91*, 12. [CrossRef]
21. Li, W.; Zhou, W.; Bai, Y.; Pickett, S.T.A.; Han, L. The smart growth of Chinese cities: Opportunities offered by vacant land. *Land Degrad. Dev.* **2018**, *29*, 3512–3520. [CrossRef]
22. Song, X.; Wen, M.; Shen, Y.; Feng, Q.; Xiang, J.; Zhang, W.; Zhao, G.; Wu, Z. Urban vacant land in growing urbanization: An international review. *J. Geogr. Sci.* **2020**, *30*, 669–687. [CrossRef]
23. Li, W.; Wang, D.; Li, H.; Wang, J.; Zhu, Y.; Yang, Y. Quantifying the spatial arrangement of underutilized land in a rapidly urbanized rust belt city: The case of Changchun City. *Land Use Policy* **2019**, *83*, 113–123. [CrossRef]
24. Sperandelli, D.I.; Dupas, F.A.; Dias Pons, N.A. Dynamics of Urban Sprawl, Vacant Land, and Green Spaces on the Metropolitan Fringe of Sao Paulo, Brazil. *J. Urban Plan. Dev.* **2013**, *139*, 274–279. [CrossRef]
25. Wang, Z.; Chen, X.; Huang, N.; Yang, Y.; Wang, L.; Wang, Y. Spatial Identification and Redevelopment Evaluation of Brownfields in the Perspective of Urban Complex Ecosystems: A Case of Wuhu City, China. *Int. J. Environ. Res. Public Health* **2022**, *19*, 478. [CrossRef]
26. Song, Y.; Lyu, Y.; Qian, S.; Zhang, X.; Lin, H.; Wang, S. Identifying urban candidate brownfield sites using multi-source data: The case of Changchun City, China. *Land Use Policy* **2022**, *117*, 106084. [CrossRef]
27. Mao, L.; Zheng, Z.; Meng, X.; Zhou, Y.; Zhao, P.; Yang, Z.; Long, Y. Large-scale automatic identification of urban vacant land using semantic segmentation of high-resolution remote sensing images. *Landsc. Urban Plan.* **2022**, *222*, 104384. [CrossRef]
28. Long, J.; Shelhamer, E.; Darrell, T.; IEEE. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. [CrossRef]
29. Liu, Y.; Duc Minh, N.; Deligiannis, N.; Ding, W.; Munteanu, A. Hourglass-ShapeNetwork Based Semantic Segmentation for High Resolution Aerial Imagery. *Remote Sens.* **2017**, *9*, 522. [CrossRef]
30. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 5–9 October 2015; pp. 234–241. [CrossRef]
31. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
32. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
33. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 833–851. [CrossRef]
34. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3349–3364. [CrossRef] [PubMed]
35. Mao, K.; Tang, H.; Zhou, Q.; Chen, Z.; Chen, Y.; Qin, Z. Retrieving land surface temperature from MODIS data by using radiance transfer equation. *J. Lanzhou Univ.* **2007**, *43*, 12.
36. Zhigan, Y. Most of China's Small and Medium-sized Lead and Zinc Smelt- ing Factories Have Ceased Production. *China Nonferrous Met. Mon.* **2009**, *1*, 2–4.
37. Binan Steel Factory Shut down after Explosions. (2013, 2013/08/09/). Philippine Daily Inquirer. Available online: https://link.gale.com/apps/doc/A339193536/GBIB?u=tsinghua&sid=bookmark-GBIB&xid=bc6b6740 (accessed on 6 September 2023).
38. Pu, Y.; Yu, H. ResUnet: A Fully Convolutional Network for Speech Enhancement in Industrial Robots. In Proceedings of the 35th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE), Kitakyushu, Japan, 19–22 July 2022; pp. 42–50. [CrossRef]
39. Ye, H.-J.; Hu, H.; Zhan, D.-C.; Sha, F. Few-shot learning via embedding adaptation with set-to-set functions. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8808–8817.
40. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
41. Wang, T.C.; Liu, M.Y.; Zhu, J.Y.; Tao, A.; Kautz, J.; Catanzaro, B. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

remote sensing

MDPI

Article

Semantic Segmentation of Urban Airborne LiDAR Point Clouds Based on Fusion Attention Mechanism and Multi-Scale Features

Jingxue Wang [1,2], Huan Li [1,*], Zhenghui Xu [1] and Xiao Xie [3]

[1] School of Geomatics, Liaoning Technical University, Fuxin 123000, China; wangjingxue@lntu.edu.cn (J.W.); 471820602@lntu.edu.cn (Z.X.)
[2] Collaborative Innovation Institute of Geospatial Information Service, Liaoning Technical University, Fuxin 123000, China
[3] Key Laboratory for Environment Computation & Sustainability of Liaoning Province, Institute of Applied Ecology, Chinese Academy of Sciences, Shenyang 110016, China; xiexiao@iae.ac.cn
* Correspondence: 472120806@stu.lntu.edu.cn

Abstract: Semantic segmentation of point clouds provided by airborne LiDAR survey in urban scenes is a great challenge. This is due to the fact that point clouds at boundaries of different types of objects are easy to be mixed and have geometric spatial similarity. In addition, the 3D descriptions of the same type of objects have different scales. To address above problems, a fusion attention convolutional network (SMAnet) was proposed in this study. The fusion attention module includes a self-attention module (SAM) and multi-head attention module (MAM). The SAM can capture feature information according to correlation of adjacent point cloud and it can distinguish the mixed point clouds with similar geometric features effectively. The MAM strengthens connections among point clouds according to different subspace features, which is beneficial for distinguishing point clouds at different scales. In feature extraction, lightweight multi-scale feature extraction layers are used to effectively utilize local information of different neighbor fields. Additionally, in order to solve the feature externalization problem and expand the network receptive field, the SoftMax-stochastic pooling (SSP) algorithm is proposed to extract global features. The ISPRS 3D Semantic Labeling Contest dataset was chosen in this study for point cloud segmentation experimentation. Results showed that the overall accuracy and average F1-score of SMAnet reach 85.7% and 75.1%, respectively. It is therefore superior to common algorithms at present. The proposed model also achieved good results on the GML(B) dataset, which proves that the model has good generalization ability.

Keywords: urban scenes; airborne LiDAR point clouds; fusion attention mechanism; multi-scale features; semantic segmentation

1. Introduction

Airborne LiDAR point clouds, scanned by light detection and ranging equipment mounted on aerial platforms, are a collection of points with original geometric properties. With the rapid development of computer vision and remote sensing technology, the application of airborne LiDAR point cloud data to urban scenes is paid more and more attention, especially in the fields of navigational positioning, automatic driving, smart city, and 3D vision [1], etc. Point clouds in urban scenes are important information carriers, which are consisted of complex surface features. In order to accurately understand 3D urban scenes from the point level, the concept of point cloud semantic segmentation was proposed. Semantic segmentation, as an important technique for LiDAR point cloud data processing, is aimed at subdividing point clouds into several specific point sets with independent attributes, recognizing the target types of point sets, and making semantic marking [2]. Semantic segmentation of airborne LiDAR point clouds in urban scene can quickly extract

typical feature information and understand complex urban scenes, so as to effectively reflect the spatial layout, development scale and greening level of the city, which has a crucial role in the fields of urban development planning, smart city and geo-database [3]. Nevertheless, semantic segmentation of point clouds is a great challenge since airborne LiDAR point clouds have characteristics of high redundancy, incompleteness and complexity [4,5].

To extract surface features from 3D point clouds, traditional methods usually construct the corresponding segmentation model according to geometric attributes and data statistical features chosen manually, such as support vector machine (SVM) [6], random forest (RF) [7], conditional random field (CRF) [8], Markov random field (MRF) [9], etc. However, selection of statistical features mainly relies on priori knowledge of operators, which has great randomness, limited ability in feature extraction of point clouds, and poor generalization. With the improvement of calculation power of computers and continuous emerging of 3D scene dataset, deep learning is taking a dominant role in the field of point cloud semantic segmentation field.

Deep learning [10] firstly was used for semantic segmentation of point clouds through rasterization of point clouds. Su et al. [11] proposed Multi-View Convolutional Neural Network (MVCNN), which got the segmentation results through convolution and aggregation of 2D images of point clouds under different perspectives. According to existing snapshots, Boulch et al. [12] produced pairs of snapshots which contained RGB views and depth maps of geometric features, then provided labels for corresponding pixels of each pair of snapshots, and then mapped the marked pixels onto the original data. Wu et al. [13] extracted features from projected 2D images by using CNN, output the pixel-by-pixel labeling chart, refined it with the conditional random field (CRF) model, and finally got the instance-level labels through the traditional clustering algorithm. Besides, voxelization of irregular 3D point clouds is a common method that researchers are used to process the original point clouds. Maturana et al. [14] proposed VoxNet network based on voxelization of point clouds, which classified point clouds by using the supervised 3D convolutional neutral network (CNN). Tchapmi et al. [15] generated the bold voxel labels through the 3D fully convolutional neural network based on voxelization of point clouds and then enhanced the prediction results by combining the trilinear interpolation and fully-connected CRF learning fine granularity. Wang et al. [16] implemented multi-scale voxelization of point clouds and extracts features, made adaptive learning of local geometric features, and realized global optimization of prediction class probabilities by using CRF with full considerations to spatial consistency of point clouds. The above semantic segmentation methods based on multi-views or voxels solve the structural problems and have some practicability. However, semantic segmentation methods based on multi-views are inevitable to lose 3D space information in the rasterization process of point clouds. The semantic segmentation methods based on voxels increase the spatial complexity and incur great expenses for storage and operation.

Therefore, some effective frameworks for direct processing of point cloud data are proposed. Qi et al. [17] designed PointNet, which made pointwise coding through multilayer perception (Mlp) and got global features through aggregation function. Nevertheless, it ignores the concept of local space and lacks extraction and utilization of local features. Qi et al. [18] proposed the improved version of PointNet, denoted as PointNet++. It proposes the density adaptive cut-in layer, learns features of point sets at different scales according to multi-layer sampling and grouping, and captures local detail information. However, PointNet++ still processes each point independently, without considerations to connections among neighbor points. In PointNet++, K nearest neighbor searching results have a problem of single direction. Jiang et al. [19] designed a scale perception descriptor for ordered coding of information from different directions and effective capture of local information of point clouds. Based on KNN construction of local neighbor graph, Wang et al. [20] used EdgeConv module to capture local geometric features of point clouds and learn features by making full use of point neighborhood information. Based on the local neighborhood processing of PointNet++, Zhao et al. [21] increased the adaptive feature adjustment module

to transform and aggregate upper and bottom information, then integrated information of different channels through Mlp and max pooling, and strengthened the description ability of features to local neighborhood. Xie et al. [22] proposed selection, aggregation, and transformation of key components by building shape context kernels, captured and spread local and global information to express internal attributes of object points. The transformation component is configured according to the overall network of PointNet. Landrieu et al. [23] divided point clouds into several super-points according to geometric shapes, and then learnt features at each super-point by using the sharing PointNet, thus enabling to predict semantic labels. Li et al. [24] proposed the X-Conv operator based on the spatial local correlation of point cloud data. The X-Conv operator standardizes the disordered point clouds through weighting and replacement of input points, and then extracts local features by using CNN. Based on the SA module of PointNet++, Qian et al. [25] introduced in the InvResMLP module to realize the high-efficiency and practical scaling of model, which solved the problem of gradient disappearance and improves ability of feature extraction. Hua et al. [26] determined features of each point through the pointwise convolution and thereby realized semantic segmentation. Hu et al. [27] replaced farthest point sampling (FPS) of PointNet++ by the random sampling and increased the perception field of each 3D point gradually through the local feature aggregation module, thus retaining the geometric details effectively. Nong et al. [28] performed densely connected the point pairs based on PointNet++, supplemented center point features to learn contextual information, and proposed an interpolation method with adaptive elevation weights to propagate point features. However, the method is limited by the lack of global information connection. Due to the great success of the transformer model [29] in capturing contextual information, researchers have introduced it into 3D point cloud processing [30]. Li et al. [31] proposed geometry-aware convolution to handle a large number of geometric instances, and then supplemented the receptive field with dense hierarchical architecture, and designed an elevation-attention module to improve the classification refinement. Zhao et al. [32] used the transformer to exchange local feature information and fit geometric spatial layout. Guo et al. [33] proposed the offset-attention module to better understand point clouds features and capture local geometric information using neighbor embedding strategy. Zhang et al. [34] introduced a bias based on the transformer model to extract relationships between local points to address the sparsity of point cloud data, and proposed a standardization set abstraction module to extract global information to complement topological relationships.

Although the above methods have achieved some progresses in semantic segmentation of point clouds, they have not adequately considered relations among point features and lack of deep interaction relations. The semantic segmentation of urban LiDAR point clouds is a challenge due to the uneven spatial data distribution of airborne laser point clouds, mixed distribution of point clouds at neighborhood surface boundary, and different scales of the objects with the same semantics. To address these problems, a convolutional network based on fusion attention mechanism which is used for 3D point clouds directly was designed in this study on the basis of PointNet++, which was called as SMAnet. Fusion attention mechanism makes parallel treatment based on self-attention mechanism (SAM) [35] and multi-head attention mechanism (MAM) [29]. The essence of SAM is to calculate similarity according to global features of each point and allocate different weights. With full considerations to interaction among points, the SAM can distinguish the mixed point clouds at surface boundary effectively. With considerations to influences of correlations of different local features on points, the multi-head attention module (MAM) was introduced in. The specific idea behind MAM is to divide high-dimensional features of points into different feature subspaces which contain different attribute information of points. Later, it judges feature similarity among different feature subspaces, thus adjusting subspace channel information. The MAM captures connections among different aspects of point features, makes full use of information correlation in local features, increases fine granularity of network, and can recognize surface points at different scales effectively.

The network also uses the light multi-scale feature extraction and supplements local geometric information by local features at different levels. Moreover, different from previous global feature extraction based on aggregation function, a global information extraction method based on SoftMax-stochastic pooling (SSP) was designed, which expanded the receptive field of network model and increases calculation efficiency as well as segmentation accuracy.

The remainder of this study is organized as follows. Section 2 introduces the proposed SMAnet method and principle. Section 3 introduces the experiment details and experimental results. Section 4 presents the discussion including comparative analysis, ablation experiments and some other additional experiments. Section 5 summarizes experimental conclusions.

2. Methods

2.1. Introduction to the Network Structure

The overall framework of the proposed SMAnet is shown in Figure 1. It covers five modules, namely, data preprocessing layer, feature extraction layer, fusion attention layer, SSP aggregation layer and feature propagation upsampling layer. Each module is introduced as follows:

(1) Firstly, the original data were preprocessed by the grid sampling strategy with considerations to uneven distribution and density of urban LiDAR point clouds, thus getting the standardized massive point cloud dataset as the input of the feature extraction layer. The framework takes the description of point clouds in a zone (denoted as (N, D)) for example, where N is the number of points in the zone and D is the number of point cloud features.

(2) The attribute features of original point clouds cannot be classified and the original features have to be mapped into a high-dimensional space to learn the high-level semantic information. The feature extraction layer learns features of point clouds by the multi-scale and multi-level method, and it contains the sampling and grouping (SG) and CNN Block. The SG layer firstly makes uniform sampling of input point clouds and the sampling points are used as centroids. Later, the input point clouds are divided into point cloud sets of different scales according to number of points searched within different radii. The numbers of sampling centroids at three layers are $N/4$, $N/16$, and $N/64$, respectively. The numbers of searched points at different scales are denoted as S_1 and S_2. Finally, multilayer perception (Mlp) is used in CNN Block to extract features of point set in the local neighborhood. The output channel parameters of Mlp and output features of each block are shown in Figure 1a. Different from the complicated structure of PointNet++ feature extraction layer, the proposed SMAnet model applies three-layer feature extraction and takes calculation efficiency and segmentation accuracy of the model into account.

(3) To address insufficient interaction information of point clouds in PointNet++, the fusion attention layer was designed after the feature extraction layer. High-dimensional feature information was strengthened by integration SAM and MAM. The basic principle is shown in Figure 1b,c. The color intensity of segments between two points represents the strength of relations and associations of multiple aspects are expressed by combination of different colors. Some points $\left\{P_1^s, P_2^s, P_3^s, P_4^s, P_5^s, P_6^s\right\}$ are given, where P_1^s is the middle point. The SAM module adds the connection between each point and the central point through global features. In other words, a thrust was applied on the point cloud feature space to push surrounding points of feature deviation P_1^s to P_1^s and establish the relationship between surrounding points and P_1^s. Based on the diversity principle of point cloud, the MAM module explores the deep association among point cloud features in feature spaces according to correlations among different subspace features. Essentially, it applies several different forces onto P_1^s to establish multiple aspects of relations with surrounding points and associations of point clouds are simulated from different perspectives. The fusion attention layer

establishes associations among points from two aspects, thus improving of semantic segmentation accuracy of point clouds.

(4) For giving high-dimensional features of attention, max pooling will lose many important features and it cannot extract global information effectively. Hence, a new aggregation function of SSP was designed as the pooling layer and point cloud features with complicated information were aggregated selectively according to probability after smoothing of SoftMax function to extract global features and filter redundant information.

(5) The feature spreading upsampling layer and feature extraction layer both contain three layers, respectively. Features of all input points were retrieved through skip connections between the learned features and the features from the corresponding feature extraction layer. Finally, pointwise classification was carried out according to features, thus getting the semantic segmentation results.

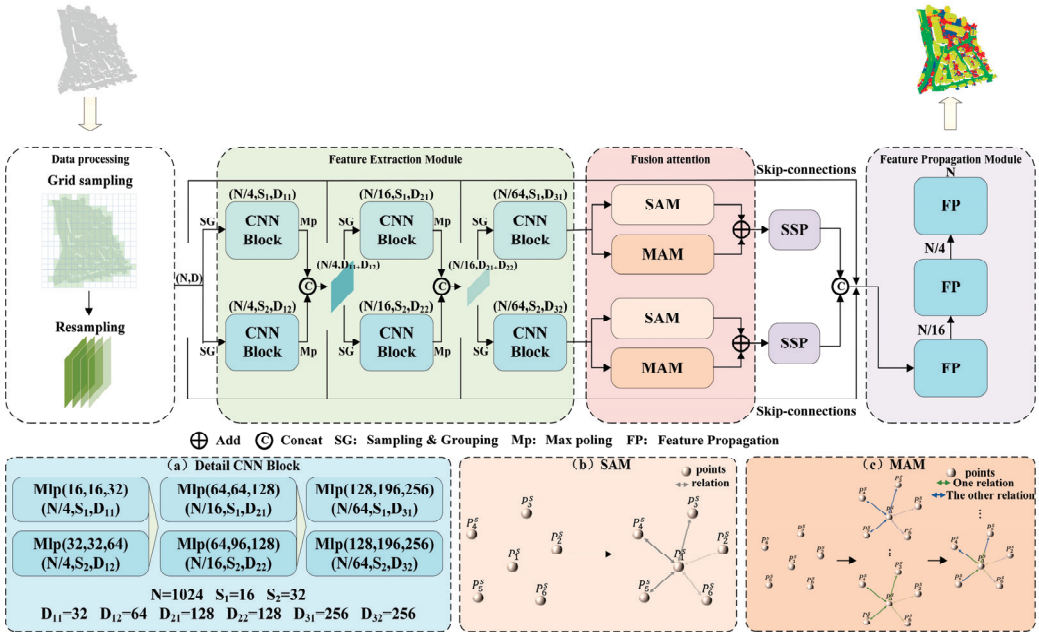


Figure 1. Framework of SMAnet model.

2.2. Data Preprocessing

Since the urban airborne LiDAR point clouds are sparse and uneven, pre-processing of the raw point cloud data is required. To protect completeness of surface features as much as possible and decrease influences of surface feature size on classification accuracy, we adopted the grid sampling strategy to assure quality of input data and amplify the limited data size [36].

It can be seen from Figure 2 that given a 3D ALS point cloud set with r disordered points $\{O_i | i = 1, 2, \ldots, r\}$, $O_i \in \mathbb{R}^d$, where $d = 6$, indicating that each point contains six features, including coordinate data of point (x, y, z), laser scanning intensity, return number, number of returns. To build the training set, the maximum and minimum coordinates along x and y were extracted from the training set, and the length and width of the whole dataset block were calculated. For grid transformation the whole block, points in each grid were determined. According to the preset sampling window size (bs) and the movement step length (s), the whole grid region was retrieved by moving, where the step length was smaller than or equal to the block size. Blocks with points less than half of the preset number of sampling points were deleted. In Figure 2, the blue blocks are deleted ones and the green blocks are the retained ones. The test set was built is the same way. The sliding step length of the test set is equal to the window size and blocks without sufficient points were not deleted. In the training process, it needs enough data for feedforward, feedback, and weight updating of the deep network. To assure that weights of each training of the

deep network can be propagated very well, it has to guarantee consistency in number of point clouds of each block. Hence, resampling of points in the block is needed by using Bootstrap sampling method to complete resampling of each block in the dataset. The resampling dataset of blocks was recorded as $\left\{E_{n_c}^{v} \middle| v = 1, 2, 3 \ldots, m\right\}$, where $E_{n_c}^{v}$ is the set where has n_c points in the block v and n_c is the fixed number. Finally, the min-max normalization [37] was performed to coordinates of point (x, y, z) and laser intensity. The normalization formula is expressed as Equation (1):

$$\overline{F} = \frac{F - F_{\min}}{F_{\max} - F_{\min}} \tag{1}$$

where F can be viewed as the feature that data have to be normalized and $F \in \mathbb{R}^j$, where $j = 4$. $F_{\max}$ is the maximum value of the feature and $F_{\min}$ is the minimum value of the feature.

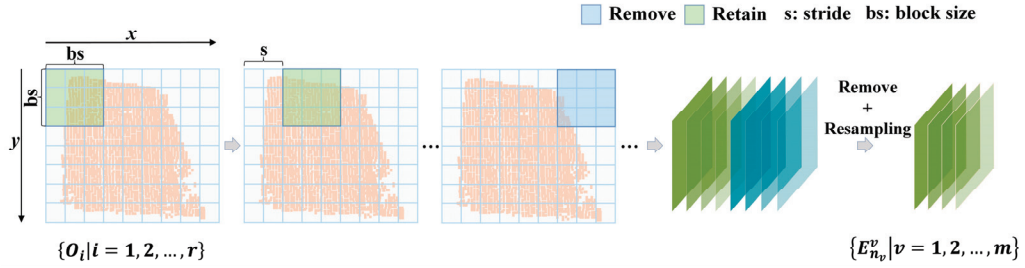


Figure 2. Grid sampling.

2.3. Feature Extraction

For semantic segmentation of point clouds, the original attribute features of point cloud data along are far insufficient. It has to learn point features of deeper layers to explore internal attributes of each point. In this study, points were aggregated into groups of different layers by using the feature extraction module and multi-scale features were extracted step by step. It is the first part of the SMAnet. The feature extraction process of each layer is shown in Figure 3. The input data are the data set P_{in} which is gained from normalization of output results (E) of data preprocessing, that is, the input point data $P_{in} = \{p_1, p_2, \ldots, p_N\}$, where P_{in} is the $N \times D$ matrix, N is the quantity of points, and D is features of point. In SG module, we firstly implemented the farthest point sampling [38] of P_{in} to determine the centroids of uniform distribution. Groups were then constructed through ball query (BQ) [18] through these centroids. The constructed groups were recorded as $P_U = \{g_1, g_2, \ldots, g_G\}$, where $g_1, g_2, \ldots, g_G$ are the set of group sampling gained by each centroid, $P_U \in \mathbb{R}^{D \times G \times S}$, G is the number of groups (that is, the number of centroids of the farthest point sampling) and S represents the number of sampling points based on each centroid.

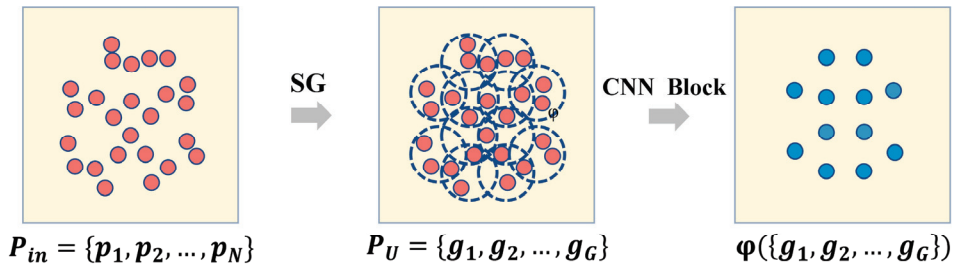


Figure 3. Feature extraction module.

Local feature extraction was performed to each centroid group (g) gained after SG sampling and grouping. P_U which was gained in the above text was input into the CNN Block module and convolution, BN [39] batch normalization and ReLU [40] nonlinear

activation functional operation were performed based on multilayer perception (Mlp) as Equation (2):

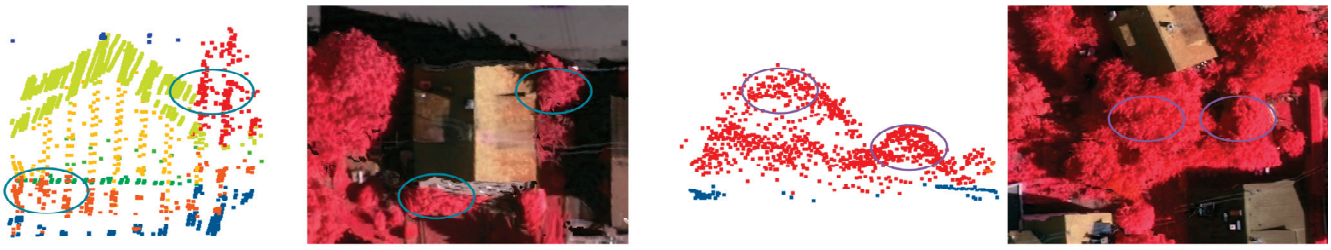
$$\varphi(g_1, g_2, \ldots, g_G) = \mathrm{Mlp}(\mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}(P_U)))) \quad (2)$$

In this study, neighborhood aggregation of centroids was carried out according to multilayer structure based on different radii. Moreover, features on different scales were combined. This process can determine local features of different scales well and supplement contextual information.

2.4. Fusion Attention Mechanism

2.4.1. Self Attention Module (SAM)

Urban airborne LiDAR point clouds are complex and diverse. During semantic segmentation of point cloud data in urban scenes, points of adjacent surface categories are easy to be confused (especially at the connection among different surface features). It can be seen from Figure 4a,b that the point clouds of this block are relatively complicated and cover several types of surface features. Point clouds of adjacent surface features were mixed and difficult to be distinguished effectively. As shown in the green circle in the figure, in one case there are roofs and trees which have height feature similarity, and in the other case there are neighbouring shrubs and facades. These adjacent point clouds of different surface objects with similar features are often classified wrongly. Although multi-scale features gained from multilayer learning can supplement local features very well and high-level expression of point clouds can be gained, the information utilization is low and classification has poor precision due to lack of associations among features. To address these problems, the SAM module was introduced into the model. This module allocates different weights to point cloud features according to their associations, thus establishing relations among global features. This improvement strengthens inner links among high-dimensional features of points, and effectively solves the problem that points in adjacent regions of different surface features are difficult to be distinguished.



(**a**) (**b**) (**c**) (**d**)

Figure 4. (**a**) is mixed region of buildings, shrubs and tree points, Tree points in region (**c**) have different scales, (**b**–**d**) are colour images of the corresponding regions of (**a**–**c**).

The specific operation of SAM module is shown in Figure 5. $\varphi(g_1, g_2, \ldots, g_G)$ after feature extraction was used as input, which was recorded as $P^{D\times G\times S}$. It was input into the convolutional layer with D output channels, getting two new feature mapping $Q^{D\times G\times S}$ and $K^{D\times G\times S}$. After reshaping of Q and K, $Q^{D\times M}$ and $K^{D\times M}$ were acquired, where $M = G \times S$, representing quantity of point clouds. The calculation formula of self-attention matrix (Sam) is expressed as Equation (3), where $Sam \in \mathbb{R}^{M\times M}$.

$$Sam_e^b = \frac{exp((Q^{\mathrm{T}}K/\sqrt{k})_e^b)}{\sum_{w=1}^{M} exp((Q^{\mathrm{T}}K/\sqrt{k})_w^b)} \quad (3)$$

where b and e are the positions of the features. $\left(Q^{\mathrm{T}}K/\sqrt{k}\right)_e^b$ is point feature at the appoint position of row b, column e. Specifically, $b = 1, 2, \ldots, M$ and $e = 1, 2, \ldots, M$. Sam_e^b refers to influences of b-th feature on e-th feature, $Sam_e^b \in Sam$. The attention score matrix Sam is essentially calculated from $Q^{\mathrm{T}}K$ dot product and SoftMax function. Two features with the higher similarity also have the stronger association. k is the number of channels with a key value of K. For great feature values, $Q^{\mathrm{T}}K$ in Equation (3) increases significantly. To prevent partial derivative approaching to 0 due to excessive input value of SoftMax [41], such effect is offset by dividing by $\sqrt{k}$ [29].

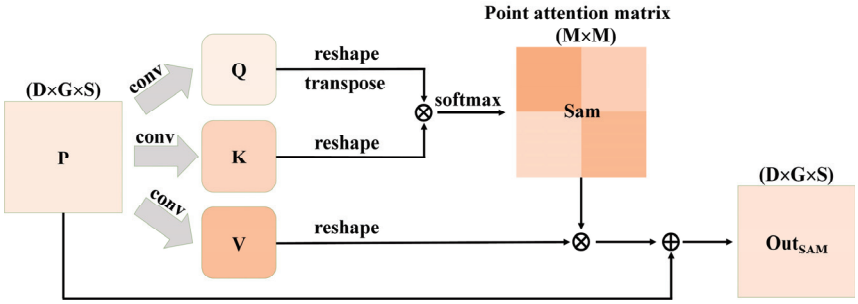


Figure 5. Operation process of SAM module.

Moreover, the input feature P was input into a convolutional layer with D channels to generate new feature mapping $V^{D\times G\times S}$. It was reshaped as $V^{D\times M}$. Subsequently, the matrixes of V and the transposition of Sam were multiplied and the result dimensional shape was adjusted as $\mathbb{R}^{D\times G\times S}$. Finally, it was multiplied with the scale parameter α and calculate sum of elements with original data P. The feature matrixes $Out_{SAM} \in \mathbb{R}^{D\times G\times S}$ with associations were output:

$$Out_{SAM} = \alpha(V \cdot Sam^{T}) + P \tag{4}$$

where α is a learnable parameter and it is initialized as 0. It makes the network firstly rely on feature information at the current position and then learn long-distance information slowly. P is the original input data and it makes the ultimately generated features have global context information. The SAM module aims to adjust the feature information for each point by allocating weights to the overall features of the point. This treatment established global dependences among different features, so that the network is beneficial for classification of point clouds in the mixing region.

2.4.2. Multi-Head Attention Module (MAM)

Similar with "the same object with different spectra" in remote-sensing images, there's also a phenomenon of "the same object with different structures" in LiDAR point clouds, especially for point clouds of trees. It can be seen from Figure 4c,d that point clouds of two trees in the oval region have different heights and geometric structures. Due to uneven scale of point clouds of the same surface feature, the point clouds of trees are often wrongly classified as other point clouds. Although SAM can use global information effectively, it will pay attention to its position excessively during coding of information of the current position [42], but uses feature information of other positions insufficiently, thus resulting in information deviation. This is disadvantageous for point cloud classification of surface features with complicated structures. In fact, local features of each point in a complicated feature space are corresponding to diversity attribute of the point. Hence, associations of several features from different aspects in the deep feature space have to be captured fully to establish associations among different points to extract effective information better. To make full use of associations among local features of points, the model is expected to learn different behaviors and capture dependences among different attributes in the feature subspaces based on local features. In this study, the MAM was introduced into processing

of point cloud data. It uses query, key, and value (QKV) together in different subspaces to supplement semantic associations among different points, thus strengthening associations in point cloud deeply. This is beneficial for classification of the same category of point clouds with different scales.

The specific process of MAM is shown in Figure 6. The input data $P^{D\times G\times S}$ were input into several convolutional layers with D channels and the channels were divided into h heads, getting feature mapping of different subspaces Q_o, K_o and V_o. Specifically, $Q_o, K_o, V_o \in \mathbb{R}^{L\times G\times S}$, where $L = D/h$ and $o = 1, 2, \ldots, h$. The calculation formula of feature mapping of each subspace is:

$$\begin{array}{l} Q_o = W_o^Q \otimes P + b_o^Q \\ K_o = W_o^K \otimes P + b_o^K \\ V_o = W_o^V \otimes P + b_o^V \end{array} \qquad o = 1, 2, 3, \ldots, h \tag{5}$$

where $W_o^Q, W_o^K, W_o^V \in \mathbb{R}^{L\times L}$, $b_o^Q, b_o^K, b_o^V \in \mathbb{R}^{L\times G\times S}$. Subsequently, the mapped features Q_o,K_o and V_o were reshaped into $Q_o^{L\times M}$, $K_o^{L\times M}$ and $V_o^{L\times M}$. Next, they were input into the scaled dot-product attention module to adjust feature channel information of different subspaces. The process of the scaled dot-product attention module is similar with that of SAM. By calculating the subspace attention Ma_o, $Ma_o \in \mathbb{R}^{M\times M}$, different subspace feature matrix after enhancement (Mam_o) was finally calculated. The specific calculation process is:

$$Mam_o(Q_o, K_o, V_o) = V_o \cdot (\text{Softmax}\left(Q_o^{\mathrm{T}} K_o / \sqrt{l}\right))^{\mathrm{T}} \qquad o = 1, 2, 3, \ldots, h \tag{6}$$

where Mam_o means to establish the associated feature matrix in each subspace, $Mam_o \in \mathbb{R}^{L\times G\times S}$, and l is the number of channels of K_o. Later, each subspace feature was combined together and then multiplied with the learning parameter (β), followed by adding the original data (P) to supplement information. Finally, the feature matrix ($Out_{MAM} \in \mathbb{R}^{D\times G\times S}$) with associations of several features was output, as shown in Equation (7).

$$Out_{MAM} = \beta \cdot \text{Concat}(Mam_o) + P \qquad o = 1, 2, 3, \ldots, h \tag{7}$$

where β is initialized as 0 and Concat expresses combination of features. The main purpose of MAM is to learn features and optimize different features of each input data by establishing different subspace attention mechanism according to multiple sets of QKVs, thus balancing possible deviations of SAM. As a result, the semantic features of point clouds have more diversified expressions and the model effect is improved better.

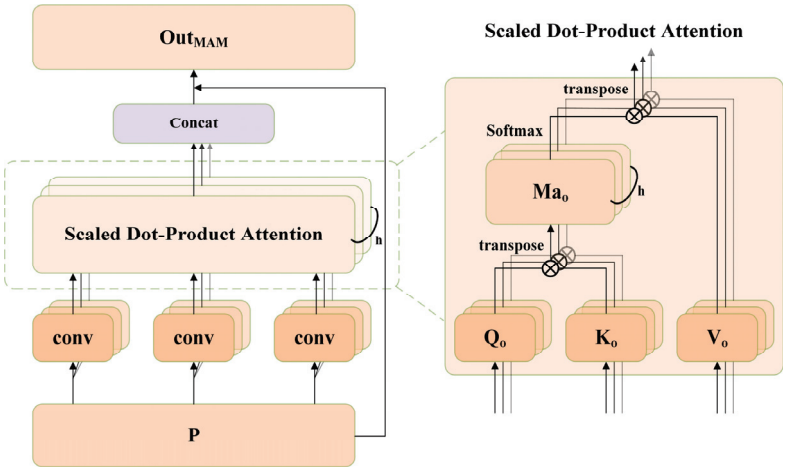


Figure 6. Operation process of MAM.

Point features extracted by multiple processes belong to high-dimensional spatial features and they contain a lot of information, all of which have their own positions and internal attributes. For better understanding and simulating the complicated interaction relationship among different points, associations of features of discrete points were established from two aspects. On one hand, the SAM analyzes global features of points and supplements the key information of global associations among points from the perspective of global features. On the other hand, the MAM interprets deep connections among discrete points from different feature spaces according to the philosophy that point features are diversified, and it considers association among discrete points more comprehensively. Additionally, the semantic information of point clouds was strengthened by the strategy of parallel SAM and MAM. In other words, association information extracted by SAM and MAM was integrated, which not only can reflect global information association, but also can reflect fusion attention feature matrix of local information associations. Subsequently, global features are extracted through aggregation function. Dimensions of features are decreased and redundant information is reduced. Finally, fusion attention feature matrixes at different scales were combined to supplement the context information.

2.5. SSP Aggregation Algorithm

After associations of features are established based on fusion attention mechanism, it has to extract global information from points with close feature associations by using aggregation function. In PointNet++, the global features are extracted through max pooling (MP) [43]. By selecting the maximum value in the pooling region through MP, the texture information can be well learnt, which has the advantage of high-efficiency memory. However, the MP is weak in information retaining and it is easy to lose key information. To explore information of deep point clouds, stochastic pooling [44] was introduced in to process point features. Stochastic pooling calculates probability in feature spaces by using the pooling window with a fixed size and chooses features randomly according to probability, which is conducive to extract information of point cloud spaces. However, extreme situations in feature spaces, for example, negative features are all 0 during function activation based on ReLU, may disturb feature information, thus bringing significant deviations in probability. As a result, it cannot extract effective information accurately. Moreover, it may lose some information upon great fluctuation of the probability space, which is also against extraction of global features. Hence, SoftMax-stochastic pooling (SSP) was designed in this study based on stochastic pooling. Essentially, the SSP normalizes probability through SoftMax function and implements smoothing the feature probability space. This aggregation function makes more information to be used, expands the receptive field of the network, and increases the information acquisition ability of the model.

The SSP aggregation algorithm is shown in Figure 7. For the input feature I, probability of each group was calculated under the fixed pooling kernel. Take the first group for example. The probability was denoted as $\{sp_u^1 | u = 1, 2, 3, \ldots, a\}$, where sp_u^1 is the probability set of the first column, and a is the quantity of features in each column. This probability is the proportion of feature values of each position in all feature values of the group. Next, smoothing of probability of each group was carried out based on SoftMax, thus getting the relatively gentle probability set $\{ssp_u^1 | u = 1, 2, 3, \ldots, a\}$. Finally, extraction of features is carried based on processed probabilities. The SSP algorithm is realized through average pooling and K-nearest neighbor sampling [38], as shown in Equations (8) and (9).

$$sp = \frac{I}{\lambda \times \delta(\mathrm{AP}(I))} \tag{8}$$

$$\mathrm{Out}_{\mathrm{SSP}} = \gamma(\mathrm{Softmax}(sp)) \tag{9}$$

where I is the output of the fusion attention module, AP refers to average pooling, δ refers to the nearest neighbor interpolation, λ means the number of K-nearest neighbor samples, and γ expresses probability-based sampling. The SSP algorithm can prevent overfitting of

the model effectively, increase perception field of features, and deepen fine granularity of the network.

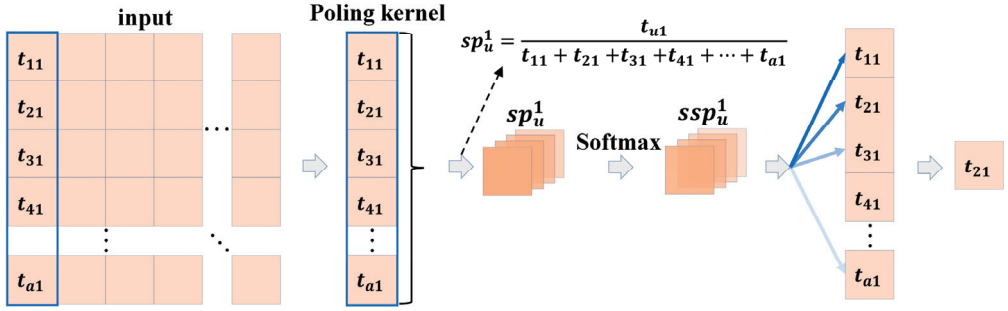


Figure 7. SSP aggregation algorithm.

2.6. Upsampling and Semantic Segmentation

Downsampling was applied in the feature extraction layer. Although it can guarantee the acquisition of global information, it may fail to complete the segmentation task due to the decrease of points. To assure the same numbers between input and output points of the model, the adjacent point features were chosen for interpolation and features were propagated from subsampling point clouds to the denser point clouds. Feature propagation (FP) [18] is mainly realized through linear interpolation and Mlp. It makes upsampling layer by layer and supplements points of high-dimensional features. Essentially, feature propagation is realized according to K-nearest neighbor (KNN) [45] and inverse distance weighted interpolation (IDW) [46]. The point closer to the interpolation point has the higher weight and greater influences on features of the interpolation point. Each FP layer combines the interpolation features and features of the corresponding feature extraction layer through skip connections to supplement information. Finally, eigenvectors were updated by Mlp.

The original point cloud features are supplemented through upsampling, which can realize semantic segmentation of point clouds. It can be seen from Figure 8 that N $\times$ 9 tensor was acquired through multi-layer convolution of significant features gained through above feature propagation. Later, predicted scores of each point were calculated from the Softmax function, and the index corresponding to the maximum score was used as the prediction result. For the training set, the prediction results and real labels were implanted into cross-entropy loss function to calculate losses, implement backpropagation and calculate gradient. Later, weights of the model were updated through the optimizer and the model parameters were optimized. In addition, the validation set is necessary to evaluate and monitor the performance of the model during the training process. In this paper, the preprocessed training set was divided into a training subset and a validation subset according to the ratio of 8:2. The hyperparameters were adjusted according to the training and validation effects, and then the adjusted hyperparameters were used to train the original training set. The model was trained from scratch until reaching the optimal solution. The above trained model was tested by using the test set. A nine-dimensional vector of each point was output according to forward propagation. This vector expresses the probability for each point belonging to each of the nine categories $c = \left\{ c_f \middle| f = 1, 2, 3 \ldots, 9 \right\}$. The index f corresponding to the maximum probability was used as the corresponding tag to be distributed to each point and used as the semantic segmentation results of each point in the test set.

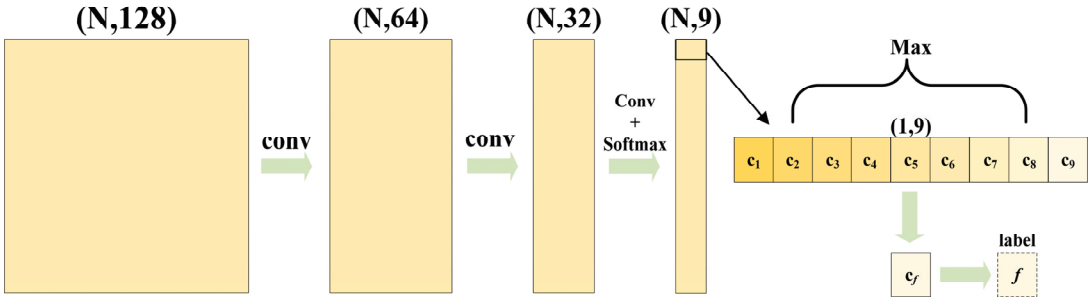
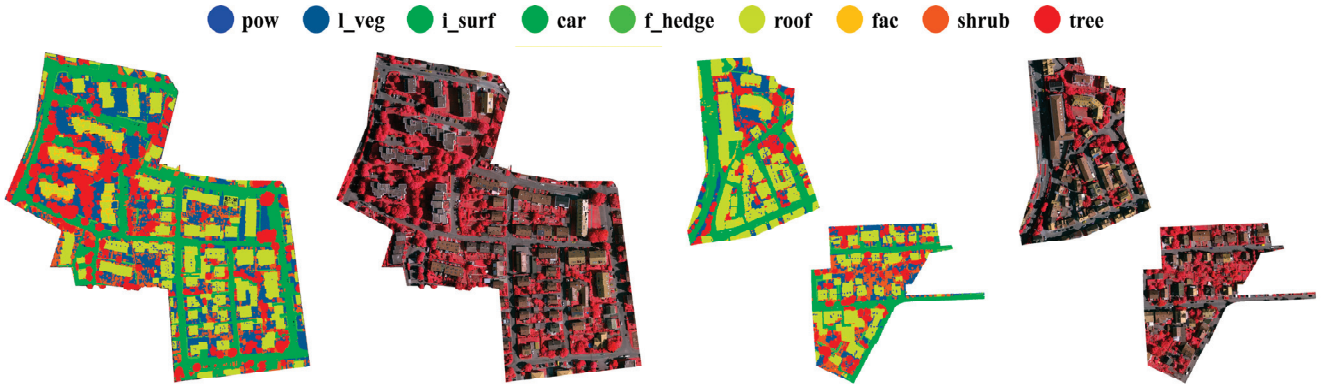


Figure 8. Semantic labeling.

3. Results

3.1. Brief Introduction to Experimental Data

The experiment used the Germany Vaihingen Town 3D Semantic Labeling Contest dataset provided by International Society for Photogrammetry and Remote Sensing (ISPRS) [47,48]. There are 753,876 and 411,722 points in the training set and test set, where the distribution of the LiDAR point cloud data and the corresponding geo-referenced IR-R-G images are shown in Figures 9a and 9b, respectively. These point cloud data were acquired based on the Leica ALS 50 system at an average altitude of 500 m above ground level. There were four points in each square meter and each laser point contains 3D point coordinates, laser intensity, return number, number of returns and corresponding semantic labels. Geo-referenced images of the entire data area have a ground sampling distance of 8 cm. The training set located in a residential area is mainly composed of buildings, vegetation and impervious surface, supplemented by few other surface features, covering an area of 399 m $\times$ 421 m. The test set is centered at Vaihingen City where has diversified surface features in dense distributions with an area of 389 m $\times$ 419 m. Unlike the training set, there are large mixed regions of shrubs and trees in addition to common surface features. The Vaihingen dataset is a typical urban surface dataset and the competition official divides surface features in the dataset into nine categories, which are powerline (pow), low vegetation (l_veg), impervious surfaces (i_surf), car, fence/hedge (f_hedge), roof, façade (fac), shrub and tree. Statistics of points in each category are shown in Table 1. Specifically, there are more points in the low vegetation, impervious surfaces, roofs, shrubs and trees categories, which are the keys of semantic segmentation. The rest surface categories with small quantities, such as powerline, car, fence/hedge, and façade, are challenges of semantic segmentation. The Vaihingen dataset contains complex and irregular surface features with rich geo-graphic environments, urban environments and buildings, which can fully validate the performance of the proposed model in urban scenes.



(**a**) Training set (**b**) Test set

Figure 9. Vaihingen dataset.

Table 1. Point quantity of different categories in Vaihingen dataset.

Class	pow	l_veg	i_surf	car	f_hedge	roof	fac	shrub	tree	Total
TraniningSet	546	180,850	193,723	4614	12,070	152,045	27,250	47,605	135,173	753,876
TestSet	600	98,690	101,986	3708	7422	109,048	11,224	24,818	54,226	411,722

3.2. Network Parameters

The proposed method was implemented using the PyTorch framework based on NVIDIA GeForce GTX1070 8G GPU. Model design parameters of the proposed method are mainly based on PointNet++ [18]. In the preprocessing of original data, the sampling block size was set 15 × 15 and 5 was chosen as the sliding step length. In each block, a total of 1024 points were chosen for sampling. There were 256, 64, and 16 centroids for group sampling in each feature extraction layer. The two radii in multi-scale BQ were [0.05, 0.1], [0.1, 0.2], and [0.2, 0.4], respectively. There were 8 heads in MAM. In model training, the batch size was set 16 and 200 epoch was trained. The Adam [49] optimizer with a learning rate of 0.001 and an attenuation step length of 10 was used. The learning rate showed exponential attenuation with the increase of iteration times.

3.3. Experimental Results and Analysis

3.3.1. Accuracy Evaluation Metrics

The ability of the proposed SMAnet was evaluated quantitatively by using common metrics for semantic segmentation of point clouds, including overall accuracy (OA), time complexity, precision, recall, and comprehensive evaluation metric (F1-score). Time complexity is the model inference time for the test set. Precision is the proportion of all points predicted by the classifier to be in the category that are accurately classified. Recall is the proportion of accurately classified points in total points of the category. F1-score is an index to measure model accuracy and it is defined as the harmonic average of precision and recall, and it is ranged between 0 and 1. OA is the proportion of accurately classified samples in total samples. The calculation formulas are shown in Equation (10).

$$\begin{aligned} P_r &= \frac{Z_{TP}}{Z_{TP}+Z_{FP}},\ R_e = \frac{Z_{TP}}{Z_{TP}+Z_{FN}} \\ F_1 &= 2 \times \frac{P_r \times R_e}{P_r + R_e} \\ Z_{OA} &= \frac{Z_{TP}+Z_{TN}}{Z_{TP}+Z_{TN}+Z_{FP}+Z_{FN}} \end{aligned} \tag{10}$$

where Z_{TP} is the number of true positives, Z_{TN} is the number of true negatives, Z_{FP} is the number of false positives, and Z_{FN} is the number of false negatives.

3.3.2. Overall Performance

The standard data for semantic segmentation of point clouds in Vaihingen is provided by ISPRS 3D Semantic Labeling Contest. With references to ISPRS 3D competition, the quantitative performance metrics, including OA, precision, recall, F1-scores and Time, were chosen to evaluate segmentation accuracy of the SMAnet model. The classification results of SMAnet and amplification results of three blocks are shown in Figure 10. The confusion matrix statistics of classification results are presented in Table 2, with the OA is 85.7% and Time is 47 s. According to analysis of Figure 10 and Table 2, SMAnet can achieve good semantic segmentation for most surface objects.

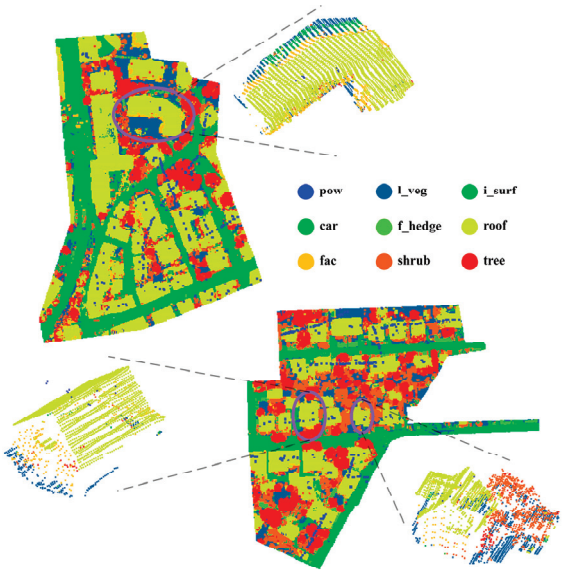


Figure 10. Experimental results of SMAnet (amplification of three local areas are shown in the circle).

Table 2. Confusion matrix of SMAnet classification results (OA = 85.7%, Time = 47 s).

Class	pow	l_veg	i_surf	car	f_hedge	roof	fac	shrub	tree
pow	**90.8**	0.0	0.0	0.5	0.0	6.7	0.5	0.2	1.3
l_veg	0.0	**83.2**	6.3	0.4	0.8	1.0	0.3	5.7	2.2
i_surf	0.0	7.2	**91.1**	0.2	0.5	0.4	0.1	0.4	0.0
car	0.0	2.2	1.8	**86.1**	1.0	2.3	1.6	4.8	0.1
f_hedge	0.0	5.7	1.1	1.4	**51.1**	1.8	1.3	21.5	16.1
roof	0.2	1.9	1.7	0.0	0.1	**93.1**	0.8	0.6	1.6
fac	0.2	5.7	0.7	0.9	1.1	10.9	**59.6**	10.8	10.2
shrub	0.0	12.4	0.6	1.7	4.3	4.9	1.6	**53.8**	20.8
tree	0.1	2.7	0.1	0.3	0.7	2.8	0.6	6.0	**86.8**
Precision	62.5	84.5	91.7	72.9	54.1	95.1	75.0	48.5	81.3
Recall	90.8	83.2	91.1	86.1	51.1	93.1	59.6	53.8	86.8
F1-score	74.1	83.9	91.4	78.9	52.5	94.1	66.4	51.0	83.9

The confusion matrix was analyzed based on the probability (recall) of correctly classified points in each category. The SMAnet model achieves the best performances in powerline, impervious surfaces and roof, with a segmentation accuracy of 90.8%, 91.1%, and 93.1%, respectively. The model can distinguish point clouds in the mixing region with similar features very well, such as powerline and roof, roof and façade, impervious surfaces and low vegetation. However, it has moderate segmentation performances to fence/hedge, façade and shrub, showing a segmentation accuracy of 51.1%, 59.6% and 53.8%, respectively. The fence/hedge is often difficult to distinguish from other categories. Specifically, 21.5% fence/hedge points are wrongly classified as shrubs, and 16.1% fence/hedge points are wrongly classified as trees. This is due to the fact that shrubs and trees have similar performances with fence/hedge in term of spatial structure, geometric features and spectral reflectance. Since façade and roof are both components of buildings, façade is easy to be mixed with adjacent shrubs and trees. Hence, façade points are mainly wrongly classified into roof, shrub and trees by 10.9%, 10.8%, and 10.2%, respectively. Shrub is mainly wrongly classified as low vegetation and trees by 12.4% and 20.8%, respectively. This is mainly caused by elevation similarity between shrub and low vegetation, and shrub has similar structural and topological relations with tree, thus resulting in the low classification precision of shrub. Besides, shrub and tree are major cause of confused semantic segmentation of most point clouds, and most surface features are wrongly classified as shrub or tree. This is due to the fact that shrub and tree are similar with most surface

features and there are mixed point clouds at boundaries. Moreover, shrub and tree have similar geometric features and they are difficult to be distinguished effectively. Although this can be improved by establishing connections among different types of points, semantic segmentation of point clouds of highly similar surface features such as fence/hedge, shrub and tree is still a great challenge.

4. Discussion

4.1. Comparative Experiments

To further verify performances of the SMAnet model, it was compared with 8 competition results provided by ISPRS 3D Semantic Labeling Contest. The IIS_7 [50] makes supervoxel segmentation of LiDAR data such as shape, color and strength, and extracts spectral and geometric features of hypervoxels by using them as the processing units. Each point is marked by KNN classifier. The UM [51] method input several features of point-like attribute information, texture information, and geometric attributes into the one-to-one classifier for semantic segmentation. The HM_1 [52] is determined by geometric features of point neighborhood and performs semantic analysis of point clouds by using RF classifier [8]. The WhuY3 [53] method transforms 3D spatial features of point clouds into 2D image features and then classifies point clouds successfully by the high-level expression of CNN extracted features. The LUH [54] designs a two-layer CRF framework and increases classification precision by iteration and context propagation of the framework. The BIJ_W [55] method proposes a pooling based on distance minimum spanning tree to process point features and increase semantic segmentation precision of point clouds. Based on the 3D coordinates and corresponding spectral information, The RIT_1 [56] method designs an end-to-end 1D fully convolutional network. The NANJ2 [57] method learns based on the deep-layer features of image context based on multi-scale CNN.

Result statistics of the proposed SMAnet model and above eight methods are listed in Table 3, mainly including F1-scores of different types of surface features, OA, average F1-score (A.F1), and Time. It can be understood from the comparative analysis that the SMAnet model shows the best general performances. The OA and average F1-scores of the SMAnet model reach 85.7% and 75.1%, which are 0.5% and 5.8% higher than those of NANJ2 method provided by the Contest. The SMAnet improves the segmentation effect of fence/hedge and façade the mostly. Its F1-scores are 11.8% and 10.1% higher compared to those of NANJ2 and LUH methods. In particular, the maximum improvement is achieved when there are limited training samples of fence/hedge and façade. Besides, SMAnet also achieves good effect in classification of powerline, car, façade and tree. NANJ2 achieves the best segmentation performances for low vegetation and shrub, and F1-scores are both 4.9% higher than that of SMAnet. This is due to the fact that the NANJ2 can learn deep-layer features better by using the multi-scale CNN. With respect to impervious surfaces, F1-scores of HM_1 and RIT_1 methods are both 0.1% higher than that of SMAnet. This is due to the fact that the HM_1 method can classify some surface points very well by using artificial features. RIT_1 extracts spectral features by using fully convolutional network (FCN), which is beneficial for extraction of Impervious surfaces. For roof, SMAnet and LUH show similar F1-scores and the F1-scores of LUH is 0.1% higher. This is due to the fact that LUH which uses voxel segmentation technique is more beneficial for semantic segmentation of point clouds of roof. Due to the model complexity, the model inference time in this paper does not reach the highest degree of the competition results. However, the SMAnet has some advantages in terms of segmentation accuracy and computational efficiency.

Table 3. Experimental results (%) of different algorithms for ISPRS 3D Semantic Labeling Contest.

Method	pow	l_veg	i_surf	car	f_hedge	roof	fac	shrub	tree	A.F1	OA	Time/s
IIS_7	54.4	65.2	85.0	57.9	28.9	90.9	-	39.5	75.6	55.3	76.2	-
UM	46.1	79.0	89.1	47.7	5.2	92.0	52.7	40.9	77.9	58.9	80.8	-
HM_1	69.8	73.8	**91.5**	58.2	29.9	91.6	54.7	47.8	80.2	66.4	80.5	120
WhuY3	37.1	81.4	90.1	63.4	23.9	93.4	47.5	39.9	78.0	61.6	82.3	70
LUH	59.6	77.5	91.1	73.1	34.0	**94.2**	56.3	46.6	83.1	68.4	81.6	-
BIJ_W	13.8	78.5	90.5	56.4	36.3	92.2	53.2	43.3	78.4	60.2	81.5	-
RIT_1	37.5	77.9	**91.5**	73.4	18.0	94.0	49.3	45.9	82.5	63.3	81.6	3.7
NANJ2	62.0	**88.8**	91.2	66.7	40.7	93.6	42.6	**55.9**	82.6	69.3	85.2	-
SMAnet	**74.1**	83.9	91.4	**78.9**	**52.5**	94.1	**66.4**	51.0	**83.9**	**75.1**	**85.7**	47

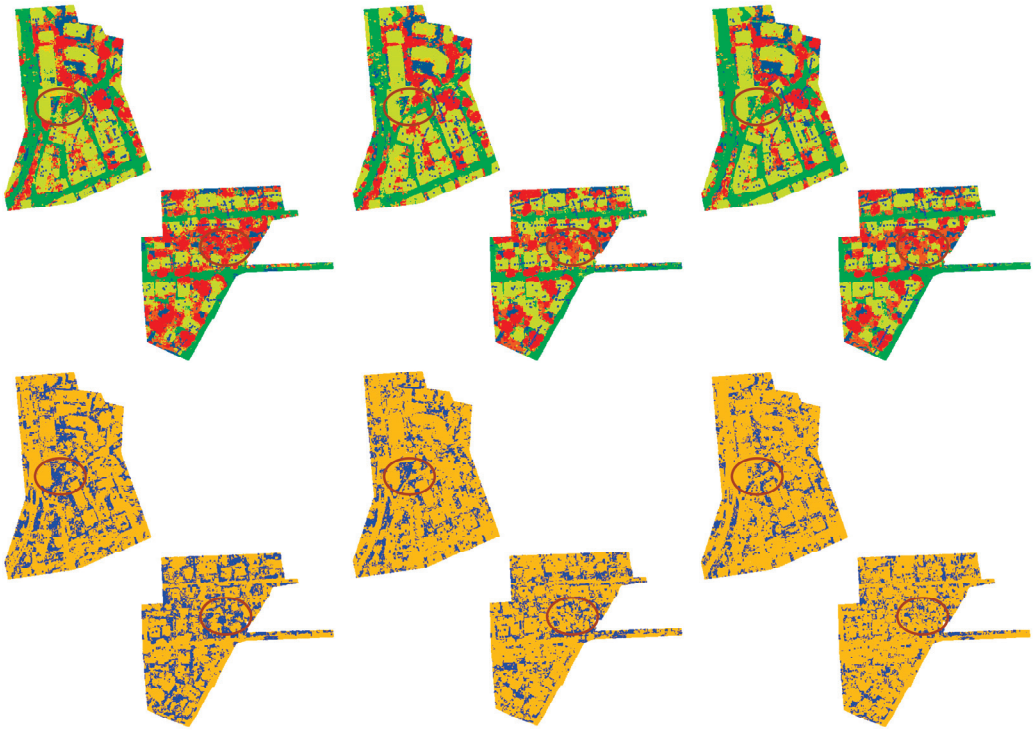
Except for comparison with the above methods, this study further compared the proposed SMAnet with PointNet [17], PointNet++ [18], PointSIFT [19], DGCNN [20], RandLANet [27], A_PointNet++ [28], GADHNet [31] and PCT [33] which have been proposed in recent years. Particularly, the DGCNN is a graph-based convolutional semantic segmentation model, the GADHNet and PCT are semantic segmentation models based on attention mechanisms. These eight models can process the original point cloud data directly, without need of rasterization of point clouds. Hence, they provide great contributions to semantic segmentation of point clouds. The results of different models are listed in Table 4. Results of different models all come from test results of open-source codes or publicly available results from papers using the same data. According to our observations, SMAnet achieves the optimal results in semantic segmentation of most surface features. Its average F1-scores and OA are 2.5% and 1.1% higher than those of the optimal reference baseline PointSIFT provided by Table 4. Moreover, the F1-scores of each surface features of the SMAnet are on average 0.4% higher than the optimal F1-scores of different surface features. For powerlines, impervious surfaces and cars, the F1 scores of A_PointNet++ are 3.5%, 0.3% and 0.3% higher than the models proposed in this paper, respectively. This is due to the fact that the former is able to distinguish surface feature points with similar elevations well by using adaptive elevation interpolation. Since the GADHNet model uses the elevation attention module to establish the point cloud elevation feature connection, it can distinguish between the uniformly distributed roof points. PointSIFT offsets insufficient neighborhood information through resampling in eight neighborhoods, thus increasing the segmentation precision of shrub by 0.1% compared to that of SMAnet. In addition, compared to other complex models, the model in this paper has low computational complexity, can inference the data quickly, and has strong practicality.

Table 4. Semantic segmentation results (%) of different models.

Method	pow	l_veg	i_surf	car	f_hedge	roof	fac	shrub	tree	A.F1	OA	Time/s
PointNet	59.9	77.5	88.7	61.1	22.9	85.0	26.9	36.6	71.4	58.9	76.2	13
DGCNN	68.1	78.2	89.1	61.6	27.9	90.5	40.8	41.4	75.5	63.6	79.7	15
PointNet++	74.3	78.9	89.7	73.9	30.6	92.1	56.8	43.3	79.7	68.8	81.5	59
RanLANet	76.8	79.3	87.3	68.1	46.7	91.0	57.9	50.6	83.7	71.2	82.2	55
A_PointNet++	**77.6**	82.7	**91.7**	**79.2**	38.9	92.2	61.3	43.2	79.1	71.8	83.5	-
PCT	75.2	80.7	90.9	70.2	41.6	91.9	61.8	48.9	82.3	71.5	83.8	42
GADHNet	75.4	82.0	91.6	77.8	44.2	**94.4**	61.5	49.6	82.6	73.2	84.5	56
PointSIFT	67.6	82.7	91.2	74.7	49.1	91.9	62.6	**51.1**	83.1	72.6	84.6	145
SMAnet	74.1	**83.9**	91.4	78.9	**52.5**	94.1	**66.4**	51.0	**83.9**	**75.1**	**85.7**	47

For visual interpretation of experimental results, two representative models of PointNet and PointNet++ were chosen. Results of PointNet and PointNet++ as well as the proposed SMAnet were displayed for qualitative analysis. PointNet is the pioneer of semantic segmentation of point clouds and PointNet++ is the improved version of PointNet.

Based on PointNet++, this study further proposed the SMAnet model. The semantic segmentation results (upper rows) and positive-negative chart (lower rows) of three models under the test set are shown in Figure 11. By comparing the results in the ellipses in the figure, experimental results of PointNet and PointNet++ have obvious wrong classifications. In particular, they have low classification precisions in the mixing distribution regions of point clouds, such as of buildings and trees, or shrubs and trees, as shown in Figure 11a,b. The SMAnet model can inhibit such wrong classification effectively and its classification precision is improved significantly compared to those of the PointNet and PointNet++ models.



(**a**) PointNet (**b**) PointNet++ (**c**) SMAnet

Figure 11. Semantic segmentation and positive-negative charts of PointNet, PointNet++ and SMAnet models, In the positive-negative chart, yellow shows points that were correctly classified and blue shows points that were incorrectly classified.

4.2. Ablation Experiments

To better understand influences of various strategies on precision of the SMAnet model, an ablation experiment was carried out in this section by adding and subtracting modules flexibly to compare abilities of different strategies. A total of six strategies were designed based on the SAM (S) module, MAM (M) module, fusion attention mechanism (SM) and softmax-based stochastic pooling (SSP). Different strategic designs are shown in Figure 12. SMAnet (BASE) has no attention module and extracts features by multi-scale sampling, and extracts global information through MP. SMAnet (S) and SMAnet (M) means to apply SAM or MAM modules based on SMAnet (BASE). SMAnet (S+M) uses different attention modules based on two scales of paths in multi-scale feature extraction. One path uses SAM module and the other path uses MAM module. SMAnet (SM) combines fusion attention mechanism and MP. SMAnet (SM+SSP) is the proposed model in this study. This mode uses fusion attention mechanism and SSP aggregation function.

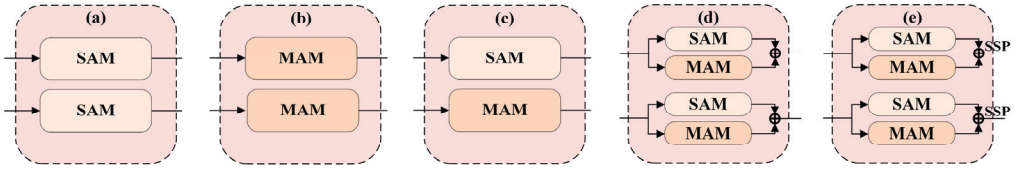


Figure 12. Schematic diagram of different strategies, (**a**) SMAnet (S), (**b**) SMAnet (M), (**c**) SMAnet (S+M), (**d**) SMAnet (SM), and (**e**) SMAnet (SM+SSP).

Statistics of experimental results of different strategies are shown in Table 5. The average F1-scores and OA of other five strategies are higher than those of SMAnet (BASE), indicating that S, M, SM modules and SSP aggregation function are all conducive to improve performances of models. SMAnet (M) has higher F1-scores than SMAnet (S) for most surface features. The F1-scores of SMAnet (M) are 9.0% and 9.4% higher in term of fence/hedge and shrub. This is due to the fact that the M module gives full considerations to deep connections of all aspects of discrete points. This is beneficial for distinguishing point clouds (e.g., fence/hedge and shrub) which have different scales and different shapes. For point clouds (e.g., powerline, car and roof) which are easy to be confused with other categories, SMAnet (S) shows better performances. Additionally, the SMAnet (SM) strategy which uses two attention modules simultaneously is superior to the model using single attention module or the SMAnet (S+M) model, showing the reasonability of the fusion attention module. In particular, F1-scores of SMAnet (SM) on most surface features are higher than those of SMAnet (S+M). This is due to the fact that SMAnet (S+M) considers only global information or subspace information in high-dimensional features, while SMAnet (SM) considers close connections among point clouds from perspectives of global information and subspace information. The average F1-scores of SMAnet (SM+SSP) are 1.3% higher than those of SMAnet (SM). This is due to the fact that SSP aggregation function can expand the receptive field of network very well, increases utilization of global information, and assures segmentation precision of the model. Although the different modules increase the computational complexity of the model, it is worth it for the increase in segmentation accuracy. To sum up, the SMAnet (SM+SSP) model has good performances in most surface features and it is applicable to semantic segmentation of point clouds well.

Table 5. Ablation experimental results (%).

Method	pow	l_veg	i_surf	car	f_hedge	roof	fac	shrub	tree	A.F1	OA	Time/s
SMAnet (BASE)	72.8	78.0	89.7	73.1	29.2	92.7	57.1	42.0	79.2	68.2	81.3	30
SMAnet (S)	77.7	78.4	89.9	78.1	33.5	93.7	60.5	41.4	80.5	70.4	82.4	34
SMAnet (M)	72.6	81.0	91.2	68.6	42.5	92.3	63.3	50.8	82.2	71.6	83.9	39
SMAnet (S+M)	68.9	81.5	91.1	73.9	48.2	92.1	63.7	55.4	82.8	73.0	84.3	35
SMAnet (SM)	69.6	82.6	91.1	74.6	51.3	93.0	64.9	54.5	82.9	73.8	84.9	43
SMAnet (SM+SSP)	74.1	83.9	91.4	78.9	52.5	94.1	66.4	51.0	83.9	75.1	85.7	47

For more intuitive observation and analysis of experimental results of different strategies, three representative areas in the test set were chosen and displayed as Figure 13a–c. They are from typical regions in the test set, as shown in oval regions in Figure 10. All three areas are in the mixing region. Semantic segmentation results of SMAnet (BASE), SMAnet (SM) and SMAnet (SM+SSP) in three areas are shown in the first three columns in Figure 13. The fourth column is the ground truth provided by ISPRS 3D official website.

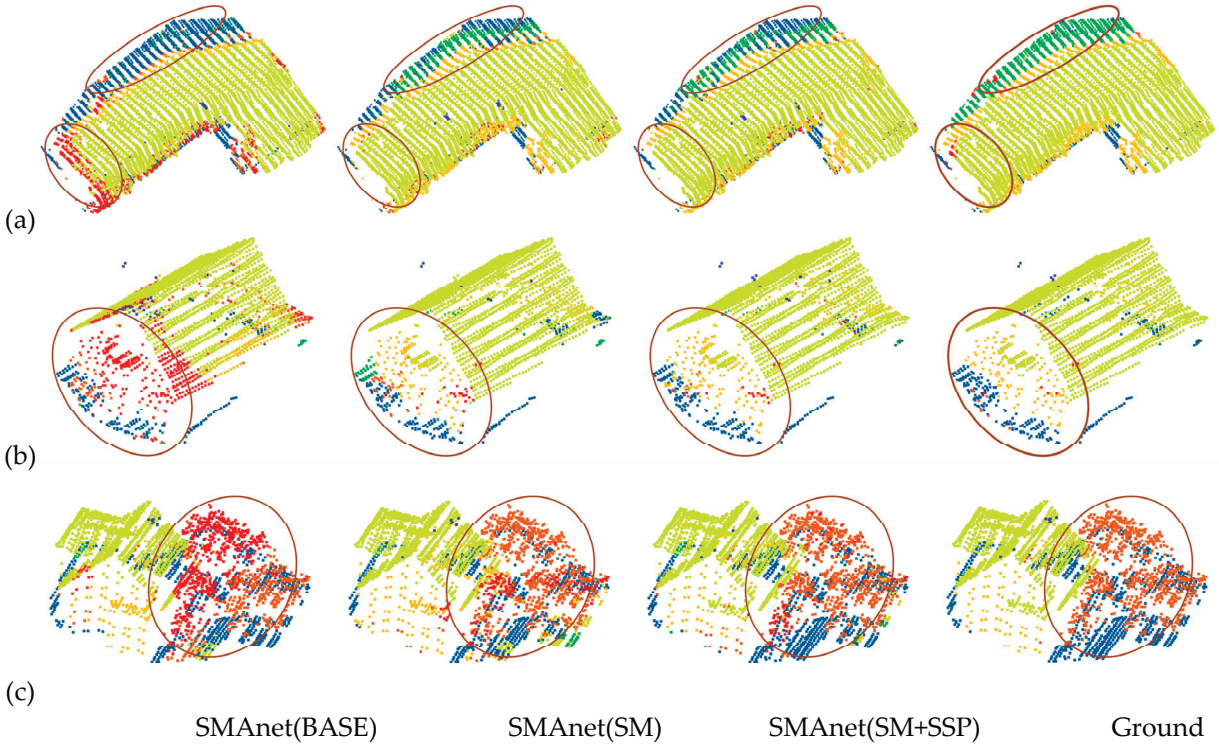


Figure 13. Classification results of SMAnet (BASE), SMAnet (SM), SMAnet (SM+SSP), and Ground Truth of test set (from left to right). Area (**a**) is the mixing region of houses with surrounding affiliated Impervious surfaces and low vegetation. Area (**b**) covers houses and low vegetation. Area (**c**) is the mixing region composed of independent houses, low vegetation and surrounding shrub and trees.

Qualitative analyses of the ablation experiments were carried out according to the elliptical circles in Figure 13. With respect to classification results in Area (a), SMAnet (BASE) achieves relatively poor classification results, while the other two strategies based on the SM module of the fusion attention mechanism can distinguish point clouds of mixed surface features very well (for example, point clouds of surface features with similar structures, such as impervious surface and low vegetation, as well as façade and tree). This is due to the fact that SM can establish the global dependence relations among features effectively. In Area (b), SMAnet (BASE) classifies most roof and façade points wrongly as tree points. SMAnet (SM+SSP) and SMAnet (SM) both can distinguish the mixed point clouds which are highly similar very well. In treatment of details (the finer points of classification), the precision of SMAnet (SM) declines since it uses MP aggregation function that may loose some important features. SMAnet (SM+SSP) can process details of point clouds in mixing regions well, since it uses SSP aggregation function that can increase information utilization very well and capture important features, thus increasing semantic segmentation precision. For the shrub point clouds of different scales in Area (c), SMAnet (BASE) is easy to classify shrub wrongly as tree, thus decreasing classification precision. SMAnet (SM+SSP) can distinguish shrubs which have complicated structures and different scales very well. In a word, SMAnet (SM+SSP) can distinguish point clouds of mixed surface features very well and shows good segmentation performances to surface features with irregular structures.

4.3. Experiments with the Number of Heads of MAM

Additionally, the following experiment was designed to verify influences of number of attention heads in MAM on capture of local feature subspace information. Based on the SMAnet model, this experiment kept the SAM module and implemented training and testing by using MAM with 2, 4, 8, and 16 heads. The results are shown in Table 6. It can be found when there are 8 heads, the SMAnet model achieved the highest average F1-scores

and OA. If there are few heads, it is easy to capture insufficient local feature subspace information. Hence, the precision increases with the increase of heads. However, excessive subspaces may affect precision [58] and inference time, bringing information redundancy and making it difficult to capture accurate information.

Table 6. Effects of head number in MAM on experimental results (%).

heads	A.F1	OA	Time/s
SMAnet (2)	70.6	83.1	36
SMAnet (4)	73.9	84.8	40
SMAnet (8)	**75.1**	**85.7**	**47**
SMAnet (16)	73.4	84.5	59

4.4. Experiments on Grid Sampling Parameters

To verify the effect of data preprocessing on the proposed model, different grid sampling strategies are selected for experiments. The experimental results are shown in Table 7, where S_Points represents the number of points sampled in the block, and P_Time represents the time of data preprocessing. It is observed that the sampling strategy with 15 × 15 block size can be well applied to the proposed SMAnet model. Large block size will lose a lot of point cloud information, leading to poor classification accuracy. The mall block size contains less information in the local area, which is not conducive to the SMAnet model establishing the feature connection between different points, and generates extra computation time.

Table 7. Experimental results (%) of different grid sampling strategies.

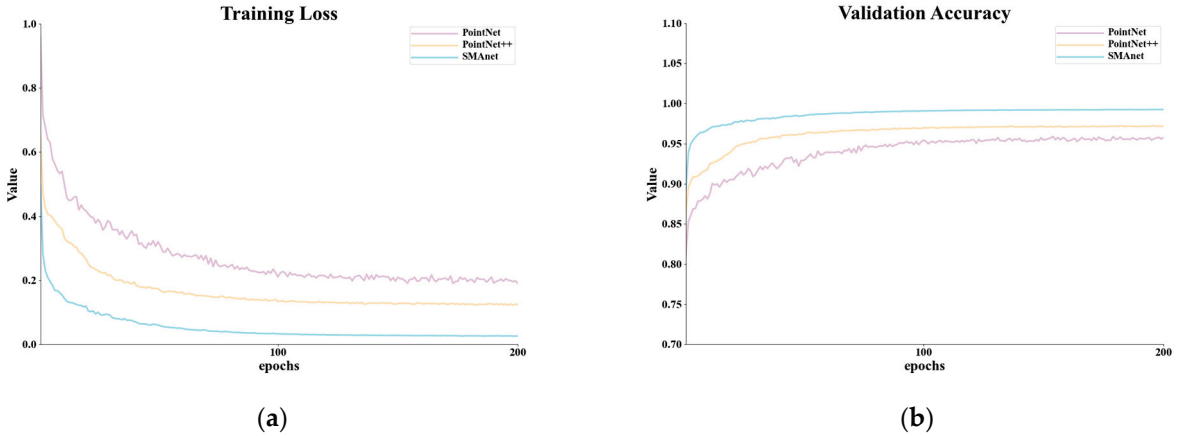
Block Size	S_Points	A. F1	OA	P_Time/s
40 m × 40 m	4096	71.1	82.2	4
30 m × 30 m	2048	71.7	82.7	5
15 m × 15 m	**1024**	**75.1**	**85.7**	**8**
10 m × 10 m	512	72.3	84.4	17

4.5. Experiments with the GML(B) Dataset

To verify the generalization ability of the SMAnet model, we performed generalization experiments on the GML(B) [59] dataset. The GML(B) dataset is acquired by an airborne Leica ALTM 2050 system, and each point only contains spatial coordinate data. It belongs to part B of the GML dataset. GML(B) dataset mainly contains four surface features: ground, buildings, high vegetation (h_ve) and low vegetation (l_ve). The experiments were conducted using the same strategy and hyperparameters as on the Vaihingen dataset. The experimental results are shown in Table 8. From Table 8, it can be seen that SMAnet has good scores in terms of inference time and classification accuracy. The proposed model can achieve better classification results for buildings, high vegetation and low vegetation points. In addition, the training loss curve and the validation accuracy curve of SMAnet are shown in Figure 14. It can be seen that the proposed model performs well, achieves a low loss, and obtains a high accuracy.

Table 8. Experimental results (%) of different models on GML(B) dataset.

Method	Ground	Buildings	h_ve	l_ve	A.F1	OA	Time/s
PointNet	99.5	76.0	88.2	42.9	76.6	96.3	39
PointNet++	99.4	80.8	90.8	47.4	79.6	96.7	145
PointSIFT	99.6	88.8	94.9	56.7	85.0	97.8	227
SMAnet	99.5	92.4	96.8	64.4	88.3	98.3	69



(**a**) (**b**)

Figure 14. (**a**) Training loss curves and (**b**) validation accuracy curves for PointNet, PointNet++ and SMAnet.

5. Conclusions

In this study, a SMAnet framework is proposed based on PointNet++. Compared with PointNet++, the SMAnet model has the following three characteristics: (1) The SMAnet model strengthens feature associations of points and supplements semantic information of each point by integrating fusion attention mechanism into the process of feature extraction. The fusion attention mechanism is composed of SAM and MAM. The SAM establishes associations through global features of points, enhances important feature channels, and overcomes difficulties in classifying mixed point clouds at boundaries of different semantic surface features. The MAM establishes internal associations of the containing features in feature subspaces of different points to explore the deep associations of point clouds. It extracts semantic associations among points more comprehensively and can classify point clouds of the same surface feature which has different scale and shapes very well. (2) SMAnet extracts multi-scale features by using the light feature learning framework and supplements local information. It considers both computational efficiency and segmentation precision. (3) The global information is extracted based on SSP. It extracts information in feature spaces selectively, expands receptive field of the network, and increases precision of semantic segmentation. The proposed model performs well on both GML(B)dataset and Vaihingen dataset.

Nevertheless, the SMAnet has the following limitations: (I) Block sampling of dataset may lose some key points. The original point cloud information will be supplemented by improved grid sampling algorithms in subsequent studies. (II) There are still poor classification precision on some surface features, such as fence/hedge and shrub. Fence/hedge and shrub are adjacent to other categories and they are difficult to distinguish effectively. It is suggested that one supplies additional relevant information such as local neighborhood geometry information in feature extraction. (III) The SSP may loose some important features since the extraction of features based on probabilities is random, which requires further constraints. In the future, the proposed model will be improved to accommodate more point cloud data, such as noisy and incomplete point clouds. Consideration is given to encoding local region point clouds through geometric and relative position information, thus complementing the neighborhood information. The attention mechanism can also be embedded into the pooling algorithm so that the network can focus on global feature extraction. In addition, reducing the computation time by designing a more lightweight framework is also a strategy to improve the performance of the model.

Author Contributions: H.L. designed the workflow and conducted the experiments, and was responsible for the main structure and writing of the paper; H.L. and J.W. discussed the results described in the paper and analyzed them; J.W., Z.X. and X.X. provided comments and suggestions on the writing of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China with Project No. 41871379, the Liaoning Revitalization Talents Program with Project No. XLYC2007026, the Fundamental Applied Research Foundation of Liaoning Province with Project No. 2022JH2/101300273 and No. 2022JH2/101300257.

Data Availability Statement: We sincerely thank the 3D Semantic Labeling Vaihingen dataset provided by International Society for Photogrammetry and Remote Sensing (ISPRS) https://www.isprs.org/education/benchmarks/UrbanSemLab/default.aspx (accessed on 23 November 2022). The GML (B) dataset URLs are available from https://github.com/bwf124565/data (accessed on 11 October 2023). The data and code presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4338–4364. [CrossRef]
2. Jing, Z.; Guan, H.; Zang, Y.; Ni, H.; Li, D.; Yu, Y. Survey of Point Cloud Semantic Segmentation Based on Deep Learning. *J. Front. Comput. Sci. Technol.* **2021**, *15*, 1–26. [CrossRef]
3. Yang, B.; Haala, N.; Dong, Z. Progress and perspectives of point cloud intelligence. *Geo-Spat. Inf. Sci.* **2023**, *26*, 189–205. [CrossRef]
4. Wahabzada, M.; Paulus, S.; Kersting, K.; Mahlein, A.K. Automated interpretation of 3D laserscanned point clouds for plant organ segmentation. *BMC Bioinform.* **2015**, *16*, 248. [CrossRef] [PubMed]
5. Grilli, E.; Menna, F.; Remondino, F. A review of point clouds segmentation and classification algorithms. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 339–344. [CrossRef]
6. Zhang, J.; Lin, X.; Ning, X. SVM-Based Classification of Segmented Airborne LiDAR Point Clouds in Urban Areas. *Remote Sens.* **2013**, *5*, 3749–3775. [CrossRef]
7. Sun, J.; Lai, Z. Airborne LiDAR Feature Selection for Urban Classification Using Random Forests. *Geomat. Inf. Sci. Wuhan Univ.* **2014**, *39*, 1310–1313. [CrossRef]
8. Zhuang, Y.; Liu, Y.; He, G.; Wang, W. Contextual classification of 3D laser points with conditional random fields in urban environments. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 3908–3913. [CrossRef]
9. Lu, Y.; Rasmussen, C. Simplified markov random fields for efficient semantic labeling of 3D point clouds. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 2690–2697. [CrossRef]
10. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
11. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-View Convolutional Neural Networks for 3D Shape Recognition. In Proceedings of the 2015 IEEE international conference on computer vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 945–953. [CrossRef]
12. Boulch, A.; Guerry, J.; Saux, B.L.; Audebert, N. SnapNet:3D point cloud semantic labeling with 2D deep segmentation networks. *Comput. Graph.* **2018**, *71*, 189–198. [CrossRef]
13. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 1887–1893. [CrossRef]
14. Maturana, D.; Scherer, S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928. [CrossRef]
15. Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. SEGCloud: Semantic Segmentation of 3D Point Clouds. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 537–547. [CrossRef]
16. Wang, L.; Huang, Y.; Shan, J.; He, L. MSNet: Multi-Scale Convolutional Network for Point Cloud Classification. *Remote Sens.* **2018**, *10*, 612. [CrossRef]
17. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85. [CrossRef]
18. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–7 December 2017; pp. 5105–5114.
19. Jiang, M.; Wu, Y.; Zhao, T.; Zhao, Z.; Lu, C. PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation. *arXiv* **2018**, arXiv:1807.00652.
20. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.* **2018**, *38*, 1–12. [CrossRef]

21. Zhao, H.; Jiang, L.; Fu, C.W.; Jia, J. PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5560–5568. [CrossRef]
22. Xie, S.; Liu, S.; Chen, Z.; Tu, Z. Attentional ShapeContextNet for Point Cloud Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4606–4615. [CrossRef]
23. Landrieu, L.; Simonovsky, M. Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4558–4567. [CrossRef]
24. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution On χ-Transformed Points. In Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018; pp. 828–838.
25. Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.; Elhoseiny, M.; Ghanem, B. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022.
26. Hua, B.S.; Tran, M.K.; Yeung, S.K. Pointwise Convolutional Neural Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 984–993.
27. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11105–11114. [CrossRef]
28. Nong, X.; Bai, W.; Liu, G. Airborne LiDAR point cloud classification using PointNet++ network with full neighborhood features. *PLoS ONE* **2023**, *18*, e0280346. [CrossRef] [PubMed]
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–7 December 2017; pp. 5998–6008.
30. Zeng, J.; Wang, D.; Chen, P. A Survey on Transformers for Point Cloud Processing: An Updated Overview. *IEEE Access* **2022**, *10*, 86510–86527. [CrossRef]
31. Li, W.; Wang, F.D.; Xia, G.S. A geometry-attentional network for ALS point cloud classification. *ISPRS J. Photogramm. Remote Sens.* **2020**, *164*, 26–40. [CrossRef]
32. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.; Koltun, V. Point Transformer. In Proceedings of the 2018 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 16259–16268. [CrossRef]
33. Guo, M.H.; Cai, J.X.; Liu, Z.N.; Mu, T.J.; Martin, R.R.; Hu, S.M. PCT: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [CrossRef]
34. Zhang, Z.; Li, T.; Tang, X.; Lei, X.; Peng, Y. Introducing Improved Transformer to Land Cover Classification Using Multispectral LiDAR Point Clouds. *Remote Sens.* **2022**, *14*, 3808. [CrossRef]
35. Nam, H.; Ha, J.W.; Kim, J. Dual Attention Networks for Multimodal Reasoning and Matching. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 299–307. [CrossRef]
36. Xu, T.; Ding, H. Deep Learning Point Cloud Classification Method Based on Fusion Graph Convolution. *Laser Optoelectron. Prog.* **2022**, *59*, 0228005. [CrossRef]
37. Jain, Y.K.; Bhandare, S.K. Min Max Normalization Based Data Perturbation Method for Privacy Protection. *Int. J. Comput. Commun. Technol.* **2013**, *2*, 45–50. [CrossRef]
38. Eldar, Y.; Lindenbaum, M.; Porat, M.; Zeevi, Y.Y. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Process.* **1997**, *6*, 1305–1315. [CrossRef]
39. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
40. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1106–1114.
41. Bouchard, G. Clustering and Classification Employing Softmax Function Including Efficient Bounds. U.S. Patent 8065246, 22 November 2011.
42. Li, Y. *Research on Website Fingerprint Identification Technology of Tor Network*; People's Public Security University of China: Beijing, China, 2022. [CrossRef]
43. Murray, N.; Perronnin, F. Generalized max pooling. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 2473–2480. [CrossRef]
44. Zeiler, M.D.; Fergus, R. Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. *arXiv* **2013**, arXiv:1301.3557.
45. Fukunaga, K.; Narendra, P.M. A Branch and Bound Algorithm for Computing k-Nearest Neighbors. *IEEE Trans. Comput.* **1975**, *C-24*, 750–753. [CrossRef]
46. Setianto, A.; Triandini, T. Comparison of kriging and inverse distance weighted (IDW) interpolation methods in lineament extraction and analysis. *J. Appl. Geol.* **2013**, *5*. [CrossRef]

47. Cramer, M. The DGPF-test on digital airborne camera evaluation-overview and test design. *Photogramm.-Fernerkund.-Geoinf.* **2010**, *2*, 73–82. [CrossRef] [PubMed]
48. Niemeyer, J.; Rottensteiner, F.; Soergel, U. Contextual classification of lidar data and building object detection in urban areas. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 152–165. [CrossRef]
49. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
50. Ramiya, A.M.; Nidamanuri, R.R.; Ramakrishnan, K. A supervoxel-based spectro-spatial approach for 3D urban point cloud labelling. *Int. J. Remote Sens.* **2016**, *37*, 4172–4200. [CrossRef]
51. Horvat, D.; Zalik, B.; Mongus, D. Context-dependent detection of non-linearly distributed points for vegetation classification in airborne LiDAR. *ISPRS J. Photogramm. Remote Sens.* **2016**, *116*, 1–14. [CrossRef]
52. Steinsiek, M.; Polewski, P.; Yao, W.; Krzystek, P. Semantische analyse von ALS-und MLS-daten in urbanen gebieten mittels conditional random fields. In Proceedings of the Wissenschaftlich-Technische Jahrestagung der DGPF, Würzburg, Germany, 7–10 March 2017; pp. 521–531.
53. Yang, Z.; Jiang, W.; Xu, B.; Zhu, Q.; Jiang, S.; Huang, W. A Convolutional Neural Network-Based 3D Semantic Labeling Method for ALS Point Clouds. *Remote Sens.* **2017**, *9*, 936. [CrossRef]
54. Niemeyer, J.; Rottensteiner, F.; Soergel, U.; Heipke, C. Hierarchical higher order crf for the classification of airborne lidar point clouds in urban areas. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2016**, *41*, 655–662. [CrossRef]
55. Wang, Z.; Zhang, L.; Zhang, L.; Li, R.; Zheng, Y.; Zhu, Z. A Deep Neural Network with Spatial Pooling (DNNSP) for 3-D Point Cloud Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4597–4604. [CrossRef]
56. Yousefhussien, M.; Kelbe, D.J.; Ientilucci, E.J.; Salvaggio, C. A multi-scale fully convolutional network for semantic labeling of 3D point clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 191–204. [CrossRef]
57. Zhao, R.; Pang, M.; Wang, J. Classifying airborne LiDAR point clouds via deep features learned by a multi-scale convolutional neural network. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 960–979. [CrossRef]
58. Michel, P.; Levy, O.; Neubig, G. Are sixteen heads really better than one? In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; pp. 14014–14024.
59. Shapovalov, R.; Velizhev, A.; Barinova, O. Non-Associative Markov Networks For 3d Point Cloud Classification. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2010**, *38*, 103–108.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

Automatic Detection and Mapping of Dolines Using U-Net Model from Orthophoto Images

Ali Polat [1], İnan Keskin [2] and Özlem Polat [3,*]

[1] Department of Planning and Risk Reduction, Provincial Directorate of Disaster and Emergency, Sivas 58000, Turkey; ali.polat@afad.gov.tr
[2] Department of Civil Engineering, Faculty of Engineering, Karabük University, Karabük 78050, Turkey; inankeskin@karabuk.edu.tr
[3] Department of Mechatronics Engineering, Faculty of Technology, Sivas Cumhuriyet University, Sivas 58140, Turkey
* Correspondence: ozlem.polat@cumhuriyet.edu.tr

Abstract: A doline is a natural closed depression formed as a result of karstification, and it is the most common landform in karst areas. These depressions damage many living areas and various engineering structures, and this type of collapse event has created natural hazards in terms of human safety, agricultural activities, and the economy. Therefore, it is important to detect dolines and reveal their properties. In this study, a solution that automatically detects dolines is proposed. The proposed model was employed in a region where many dolines are found in the northwestern part of Sivas City, Turkey. A U-Net model with transfer learning techniques was applied for this task. DenseNet121 gave the best results for the segmentation of the dolines via ResNet34, and EfficientNetB3 and DenseNet121 were used with the U-Net model. The Intersection over Union (IoU) and F-score were used as model evaluation metrics. The IoU and F-score of the DenseNet121 model were calculated as 0.78 and 0.87 for the test data, respectively. Dolines were successfully predicted for the selected test area. The results were converted into a georeferenced vector file. The doline inventory maps can be easily and quickly created using this method. The results can be used in geomorphology, susceptibility, and site selection studies. In addition, this method can be used to segment other landforms in earth science studies.

Keywords: doline segmentation; deep learning; orthophoto images; geocomputation

1. Introduction

Gypsum karsts have been observed in many countries. These landforms cover an area of approximately 5% of Turkey [1]. The largest part of the distribution of gypsum is located in the Sivas basin in Turkey. This region is one of the rare areas in the world where karstic shapes with unique features are found most prominently.

For the formation and development of karstification, rocks must exist that can dissolve in water. The more easily the rocks are dissolvable in water, the faster the development of the karst topography. The term karst topography is mostly associated with limestones. However, relatively large karstic depressions are not observed in those types of areas. Gypsum is a type of low-durable rock with a hardness of around 2.0, and it melts much more easily than limestone. Therefore, its melting rate is higher than that of limestone. Dolines are the most distinctive and common landforms of karstic lands. They have a mostly circular shape, but sometimes, they can also be in different shapes. Their long axes or diameters can vary from a few meters to a kilometer. In general terms, dolines are formed by the dissolution of gypsum at the bottom and the gradual settling that occurs on the surface. Karstic landforms are formed more quickly and easily in gypsum fields. Determining the characteristics of karst shapes is of great importance in terms of both understanding the formation processes and determining new land-use models.

The collapse of gypsum in most countries results in damage to many living areas and various engineering structures. This event creates natural dangers in terms of human safety, agricultural activities, and the economy [2–7]. In addition, it has been observed that most surface water and groundwater are enriched in sulfate by dissolving gypsum in gypsiferous areas. Therefore, the determination of gypsiferous areas and their karstification characteristics should be taken into account in national- and local-scale planning in terms of addressing natural hazards and water resources.

The rapid detection of karstic shapes and their time-dependent changes are very important parameters in terms of predicting damage in karstic lands. Mapping the dolines in an area is the basis for predicting other dolines that may occur in the future. Creating a doline inventory map is quite difficult, especially in polycarst areas [8], and this is because there are often thousands of very similar dolines on the field.

Image segmentation is commonly used to find objects and borders in images [9–13]. Semantic and instance segmentation methods have been used in image segmentation projects. In this study, we used U-Net, which is one of the semantic segmentation methods that is used to predict doline areas. With this method, the image is automatically classified by assigning all of the pixels in an image to any defined class. This method uses convolutional neural network (CNN) architecture. A U-Net convolutional network [14] is an ideal network for semantic segmentation studies. U-Net consists of two parts: an encoder and decoder. It does not need a large number of labeled data, unlike other methods, and it can learn with a small number of effective data. Additionally, with U-Net, skip connections are used to transfer information from previous convolution layers to deconvolution layers. This increases segmentation performance. For this reason, a U-Net model was preferred in detecting dolines in this paper. The U-Net model requires very few training images and yields more precise results in terms of pixel locations. Therefore, it is often preferred in earth science and other studies [15–21].

The novelty of this paper is that the dolines were detected automatically by U-Net's semantic segmentation via transfer learning. There are segmentation studies in which features such as buildings, roofs, and roads were extracted from orthophoto images. However, there have been no studies on doline segmentation. The study area has thousands of dolines, and it is almost impossible to determine all of them with classical field studies. Our model successfully predicted dolines in a randomly selected area.

Literature Review

In the literature, there have been many studies conducted with various methods, especially in similar areas using machine learning and deep learning methods on doline distribution such as the following: object detection [22–26], object classification [27], image segmentation [28–30], change detection [31], building objects mapping [32], vegetation mapping [33], and structural controls [34].

In the study conducted by Mochales et al. [22], dolines were located in the Ebro basin in northern Spain. The dolines identified were filled with alluvial sediments, agricultural soils, and urban debris. For this purpose, magnetic susceptibility measurements were used, which revealed a remarkable contrast between the host rocks and cavity fillings. Pueyo-Anchuela et al. [23] proposed a geophysical routine for use in alluvial karst regions to create an integration-based model of the many methodologies in their paper. For the detection of doline areas, the geophysical characteristics involved in each approach, as well as their solutions and uncertainties, were investigated. The suggested sequence of implemented procedures indicates a gradual rise in survey time consumption, ambiguity reduction, and improved resolution. In the study by Nahhas et al. [24], a deep learning (DL)-based building detection approach using a combination of Light Detection and Range (LiDAR) data and orthophotos was investigated. The proposed model was employed on two datasets taken from an urban region with different building types, and it was tested with 21 features and 10 features. The experimental results showed that the accuracy of the tests with size reduction was higher both in the study region and in the test region.

Hussain et al. [25] investigated the Vodose and Fluvial caves in Tarimba (Goias, Brazil) with several geophysical approaches. They determined that the findings obtained from their studies were compatible with real field conditions. They found circular and concave landforms formed by karstic processes, which are known as sinkholes or dolines. In their study, Čarni et al. [26] propose using indicator species to describe sections of several dolines and their common species to determine dolines with significant conservation value for cold-adapted species. The primary goal of their work was to classify the dolines into landform-vegetation units (LVUs) by considering essential geomorphic features and indicative plant species.

Zero-shot learning (ZSL) is a method for identifying unseen objects during the training phase that has been known to be beneficial in real-world situations. In the study performed by Pradhan et al. [27], the integration of CNN and ZSL was used as a classification and feature extraction technique to map the land cover using high-resolution orthophotos. High accuracy values were obtained in the experiments, and the effectiveness of ZSL in land cover mapping based on high-resolution photographs has been proven.

Abdollahi et al. [28] presented two novel deep convolution models for the segmentation of multiple objects from aerial photos such as buildings and roads based on the U-Net family. Their attention was directed toward buildings and road networks due to their large presence within urban regions. The presented models are called U-Net with multi-level context gate (MCG-UNet) and bidirectional ConvLSTM UNet model (BCL-UNet). Compared to U-Net and BCL-UNet, MCG-UNet increased the average F1 success by 1.85% and 1.19% for road extraction and by 6.67% and 5.11% for building extraction, respectively. Road parts extraction is of great importance in most Geographic Information System (GIS) applications. To derive the road class from orthophoto images, Abdollahi and Pradhan [29] employed an integrated technique that included segmentation and classification methods with connected component analysis. There are three steps to the approach that has been proposed. The fractured pictures were first segmented using the multi-resolution segmentation approach. Using three different classification techniques (support vector machines, decision trees, and k-nearest neighbor), the results were then split into two groups: road and non-road. Finally, connected component labeling was utilized to extract road components, and morphological processing was performed to increase performance by removing off-road parts and noises. The current methods described in the literature for road extraction result in fragmented output due to obstructions like shadows, structures, and vehicles. To address the fragmented results, Abdollahi et al. [30] presented SC-roadDeepNet, which is a deep learning-based architecture that preserves structure shape and connection. In comparison to alternative models like LinkNet, ResUNet, U-Net, and VNet, the proposed approach enhances the F1 score value by 5.49%, 4.03%, 3.42%, and 2.27%, respectively.

In the study conducted by Ghasemkhani et al. [31], the proposed model describes the conversion of bare lands into settled or developed areas. The model consists of a fuzzy system and Ordered Weight Averaging (OWA) methods together. The applied model consists of four parts: physical fitness, accessibility, neighborhood effect, and calculation of general fitness. Experiments have shown that the proposed model predicts changes with high accuracy.

Abdollahi and Pradhan [32] made studies related to the mapping of building objects from aerial images. They proposed a novel deep learning structure called MultiRes-UNet. In their network, they used MultiRes-UNet blocks and convolutional operations with skip connections. They combined semantic edge knowledge with semantic polygons. They tested their model on the roof segmentation dataset and achieved a 0.78% increase in mIoU value.

In another study, Abdollahi and Pradhan [33] studied urban vegetation mapping and described how the output of the DNN model used to categorize vegetation can be interpreted using an annotation technique known as Shapley additive explanations (SHAPs). They evaluated the accuracy of their method by mapping vegetation from aerial imagery

using spectral and textural parameters, and their results showed an overall accuracy of 94.44%.

The solution dolines form characteristic landforms that are observed on the high plateaus of the Taurus Mountains. The study by Öztürk et al. [34] concentrated on how the distribution and morphometric characteristics of the dolines in the western portion of the Central Taurus Mountains were influenced by tectonic structures, drainage patterns, and slope conditions. As a result of the research, it has been revealed that the fault and joint systems formed on the thick-bedded limestone between the thrust faults are effective on the change in the intensity of doline.

2. Material and Methods

This study consists of three phases: data preparation, analysis, prediction and mapping (Figure 1). First, an area was selected for the analysis. This selection should be balanced for the doline area and the non-doline area. The second process is the labeling of images including dolines. A binary image (mask) is created by labeling all dolines. Later, desired size patches of the image are created, and the images that have no-doline data are removed. Subsequently, classical segmentation processes were performed. These processes are dataset preparation, training, testing, and prediction. In this study, a Segmentation Models library with U-Net architecture [35] was used. This is a Python library with Neural Networks for Image Segmentation based on Keras [36] and TensorFlow [37]. The last task is converting the predicted doline data to a georeferenced vector file. The experiments were completed on a desktop that has an Intel i7 CPU, 32 GB memory and NVIDIA GeForce GTX 1070 graphics card. These processes are explained in detail below.

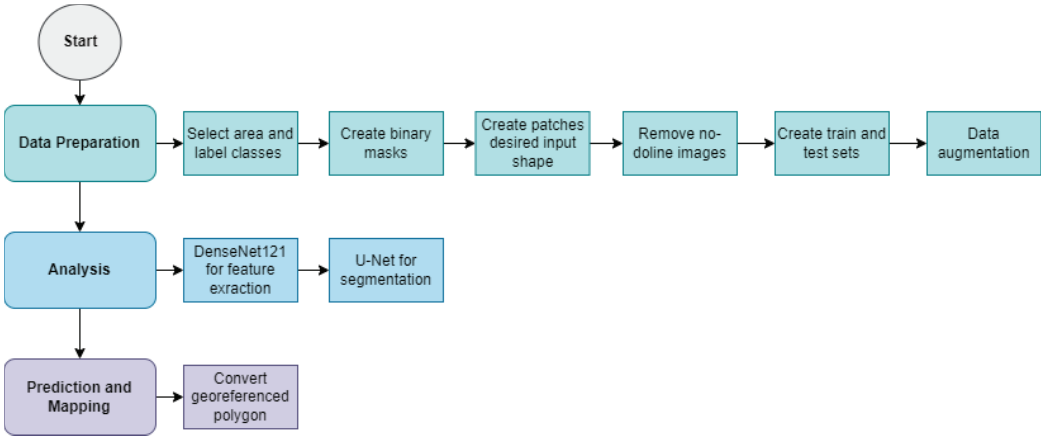


Figure 1. Workflow diagram.

2.1. Data Preparation

The data preparation process is one of the very important steps of classification and segmentation studies. Many factors in this process directly affect the results of the model. Therefore, the task of data selection and preparation of the data should be completed carefully.

In this study, orthophoto images with 0.3 m spatial resolution of the study area were used as input data. An area of about 4.835 km^2 was selected randomly to create training and test data. In the next step, boundaries of all dolines in this area were drawn and labeled. The next processes were completed by Python scripts. The first script named "create image and mask" creates two separate images in RGB and binary format. The U-Net model requires mask images related to each class. We used dolines and backgrounds as class labels. Therefore, we prepared a binary image having the values of 0 and 255. Zero values were assigned to the background pixels, and 255 values were assigned to the doline pixels. Later, 320 × 320 sized patches were created from these single images (Figure 2). Images with no doline data were removed. In addition, the number of pixels as labeled background and doline in the images was determined. Then, images containing too small

doline areas were removed. A total of 374 RGB images and the same size of mask images with the same names were created. These data were split into training and test sets. The splitting ratio was selected as 67% for training and 33% for test data. Finally, 251 training images and 123 test images were created. The splitting process was also completed for the mask images.

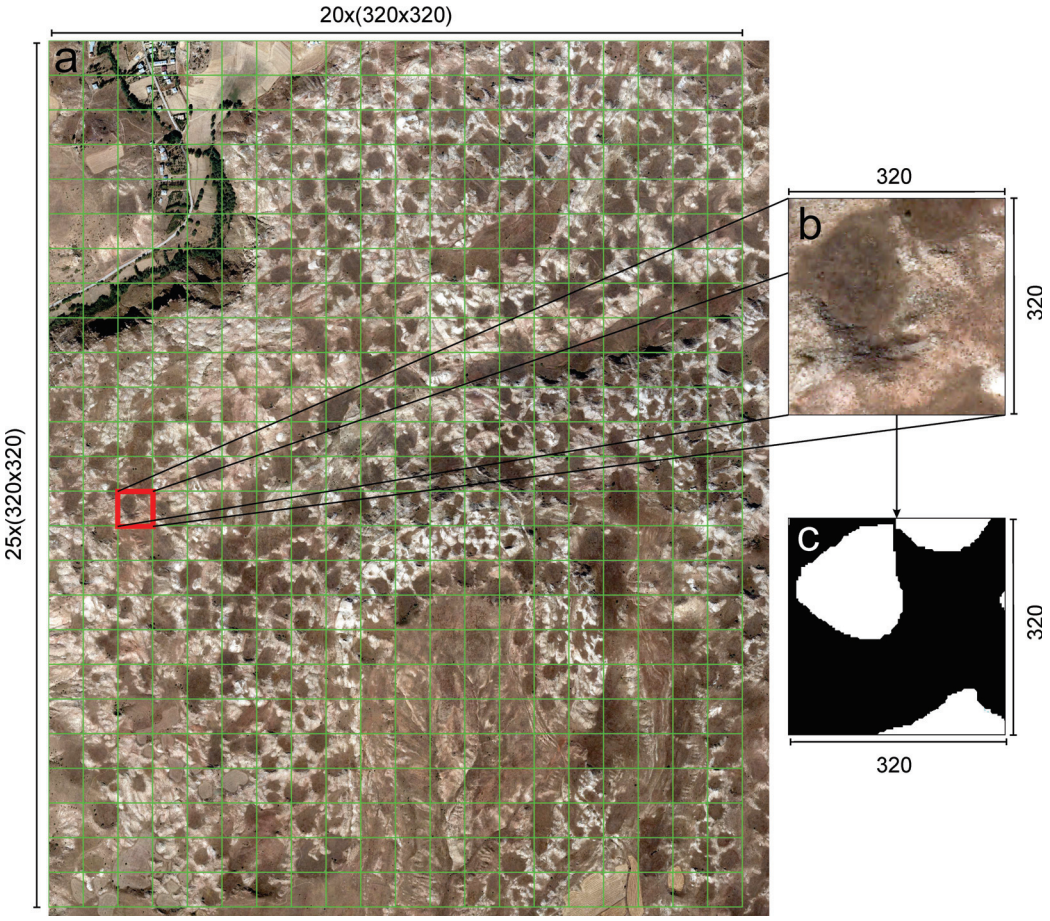


Figure 2. (**a**) Patched original image, (**b**) Sample of a patch, (**c**) Mask image of patch.

2.1.1. Study Area

The study area is located about 25 km northwest of the Sivas city (Figure 3). An area of approximately 4.835 km^2 where dolines are observed very densely has been chosen as the study area. This area consists entirely of gypsum units. The main lithology in the study area and its surroundings consists of Lower Miocene–Middle Miocene gypsum with considerable karstification (Figure 4a). There are thousands of dolines with different sizes and types within the formation. Gypsum bedding in the region is mostly not obvious. Gypsums show a massive structure with thickness ranging from 100 to 200 m [38].

The areas where dissolution dolines are frequently seen can be defined as doline karst areas. Dolines in the study region are defined as dissolution dolines with a depth of 5–20 m. These are mostly flat dolines that elongate in one direction and with shapes of circular or nearly circular. These dolines generally do not have water (Figure 4b,c).

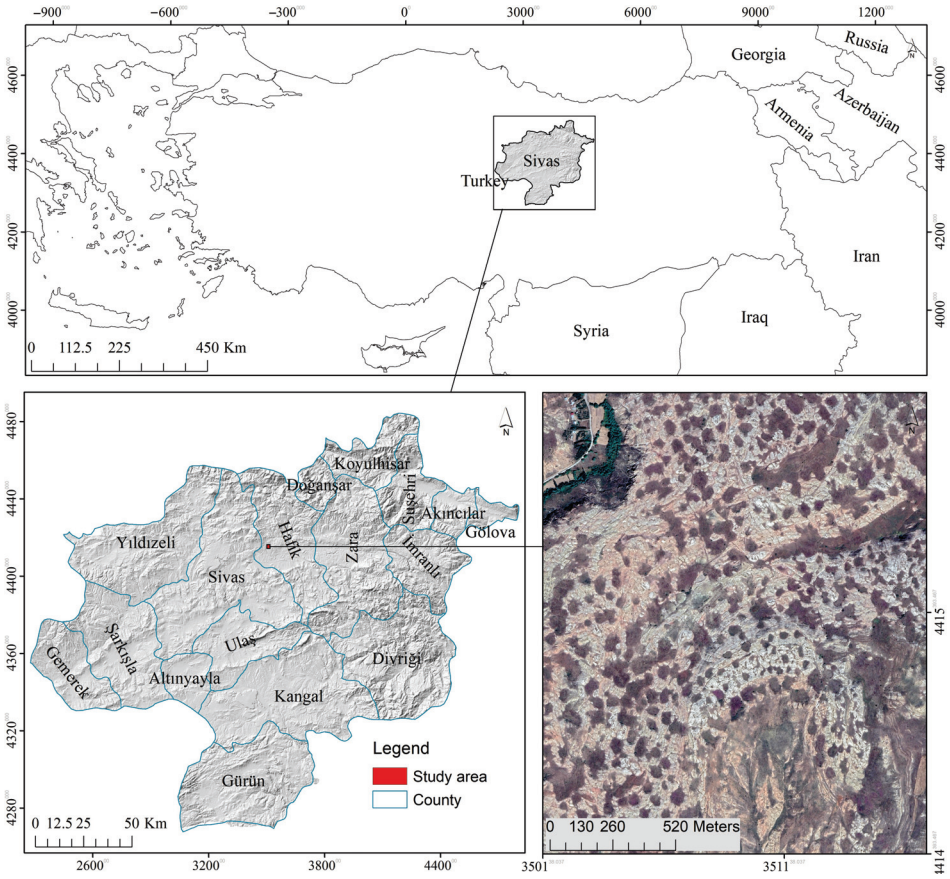


Figure 3. Location map of study area.

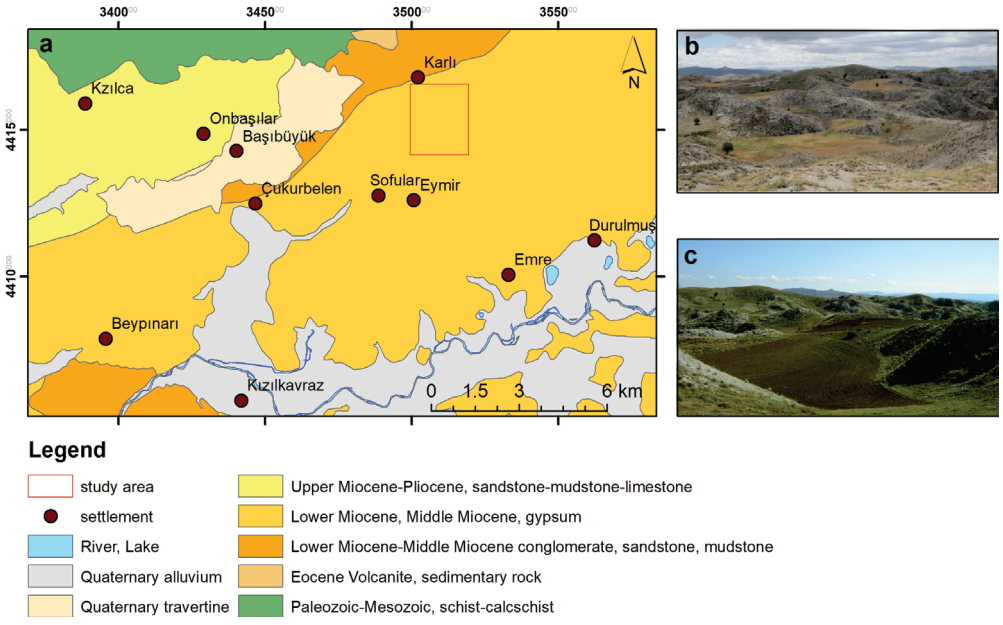


Figure 4. (**a**) Geological map of study area (1/25,000 scale geological map of General Directorate of Mineral Research and Explorations) (**b**,**c**) Doline images.

2.1.2. DenseNet121

Transfer learning is a machine learning technique. In this technique, pre-trained model weights are used for a new task. The new model is trained faster by fine-tuning using the knowledge from the previously trained models. In addition, this method provides high performance by using fewer data.

In this study, we used DenseNet121 architecture as a pre-trained model. This model consists of four dense blocks. These blocks include 6, 12, 24, and 16 convolution blocks, and each convolution block also has two convolution layers, Conv(1 × 1) and Conv(3 × 3), respectively. Also, there is a transition block between dense blocks. This block has a convolution layer, Conv(1 × 1), and a 2 × 2 average pooling layer. The last block is the classification block. This block has a global average pooling followed by the fully connected layer. There are 121 convolution layers in DenseNet architecture (Figure 5). Therefore, it is called DenseNet121. For more information, see [39].

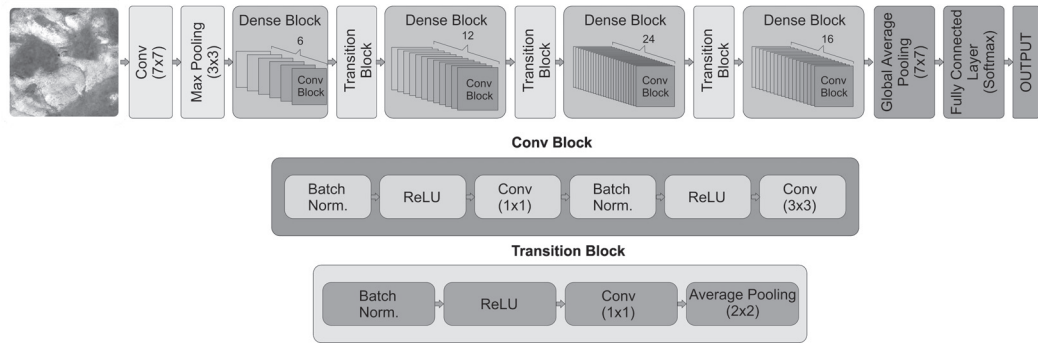


Figure 5. DenseNet121 architecture [40].

2.1.3. U-Net Model

In this study, U-Net architecture [14] developed for biomedical image segmentation was used as the segmentation model. This architecture is a type of fully convolutional networks. It consists of two parts. It is called U-Net because its architecture looks like a U-shape (Figure 6).

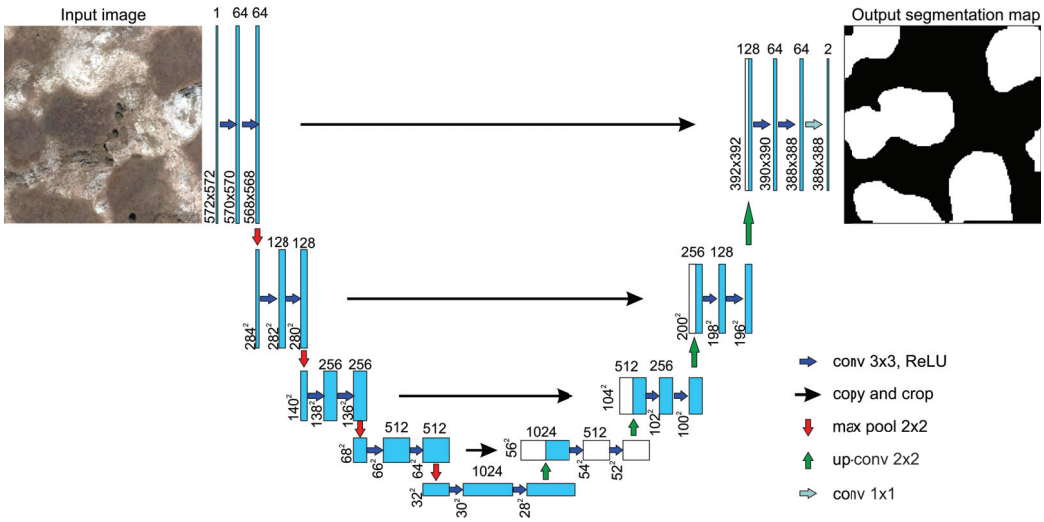


Figure 6. U-net architecture.

Conventional CNN architectures require too much training data. The U-Net model performs effectively with a limited amount of training data and produces more accurate results. The network architecture of U-Net is shown in Figure 6. It consists of a contracting path (left side) and an expanding path (right side). The contracting path captures context, and the expanding path enables precise localization. The contracting path is like the typical architecture of a convolutional network. Each standard convolution process is activated by the Rectified Linear Unit (ReLU). In the expanding path of the model, the upsampling process is applied. Thanks to the layers in the expanding path, it is aimed to augment the output resolution. The sampled output is combined with high-resolution features throughout the model for localization. Cropping is required because the boundary pixels

disappear with each convolution. In the last layer, a 1×1 convolution process is applied to transfer each feature vector to the intended number of classes.

2.1.4. Data Augmentation

The DNNs require a large amount of training data. The more data, the better the model performance. In addition, it prevents overfitting of the model. Although the U-Net architecture requires less data, we implemented a data augmentation method for our model. We used the Albumentations method [41] for data augmentation. Albumentations offers an interface for various computer vision tasks, including classification, segmentation, detection, estimation, and more. The library is extensively utilized in industry, deep learning research, machine learning competitions, and open-source projects.

Albumentations includes many transform methods, and all these transforms can be easily applied. Since we have a small dataset, we used numerous augmentation methods including:

– Horizontal flip;
– Affine and perspective transforms;
– Brightness, contrast and color adjustments;
– Sharpening and blurring process;
– Adding gaussian noise.

2.2. Evaluation Metrics and Loss Functions

In this study, IoU (Intersection over Union), also known as the Jaccard index, and F-score metrics were used to evaluate the model performance. The Jaccard index [42], also called the Jaccard similarity coefficient, is a statistical metric used to compare the similarities and differences between sample sets (1). In our study, the relationship between ground truth (real dolines) and predicted dolines was compared. The higher the Jaccard index, the higher the accuracy of the classifier.

$$J(A, B) = \frac{A \cap B}{A \cup B} \tag{1}$$

We used the F-score (Dice coefficient) as the second metric for the model evaluation. F-score is a metric that integrates both precision and recall into a single value. Precision is the proportion between the True Positives and all the Positives (2). Recall calculates what proportion of actual positives was identified correctly (3). The F-score considers the importance of precision and recall equally. The β value in Equation (4) is the F-score coefficient utilized for balancing precision and recall. The F-score value is between 0 and 1. 1 represents the highest score, and 0 represents the lowest score. The calculation of F-score is shown in Equation (4).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{2}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{3}$$

$$F_\beta(precision, recall) = (1 + \beta^2)\frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{4}$$

Segmentation model losses can be combined. We used Binary Focal Loss (BFL) and Dice loss (DL) together. BFL measures the Binary Focal Loss between the ground truth and the prediction by the following equation.

$$L(gt, pr) = -gt\alpha(1 - pr)^\gamma log(pr) - (1 - gt)\alpha pr^\gamma log(1 - pr) \tag{5}$$

where gt is ground truth, pr is prediction, α is the same as a weighting factor in balanced cross-entropy, and γ is the focusing parameter for the modulating factor.

The other loss function is DL, which is developed based on the dice coefficient for binary data and can be calculated as follows:

$$L(precision, recall) = 1 - (1 + \beta^2)\frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \quad (6)$$

where β is the coefficient for precision and recall balance.

3. Results and Discussion

In this study, the Segmentation Models library was used for the segmentation process with transfer learning techniques. The transfer learning allows you to take pre-trained model weights for any task and reuse them for another task. In this technique, the layers and weights of a pre-trained model for classification purposes are used in the first part (encoder) of the U-Net architecture. Then, the layers in the second part of the U-Net (decoder) are trained with the augmented dataset. After preparing the data, the model was trained in U-Net architecture with three different pre-trained models. These are ResNet34, EfficientNetB3 and DenseNet121. Each model was trained and evaluated using the same parameters. Model results are close to each other. However, the results of DenseNet121 have higher values than the others. Therefore, the DenseNet121 model was used for the prediction of dolines.

A new area (not including the train and test data) was selected from the study area, and the model was run. This area is about 7.289 km^2 (Figure 7). A Python script was written for the prediction task. This script loads the model, patches the image, makes a prediction, and creates a polygon file including dolines. Our script works successfully and is completed in a short time. This process was completed in approximately 30 s. A total of 808 dolines were predicted in the selected area. The total area of the dolines is 2.689 km^2. This area corresponds to 37% of the total area. The smallest doline area is 1.5 m^2, and the largest doline area is 115,775.2 m^2. All the predicted doline areas were almost the same as the real shapes of dolines. The boundaries of dolines are correctly predicted as seen in Figure 7b. Figure 7a shows the predicted dolines, and Figure 7b shows a zoomed view of an area of the image.

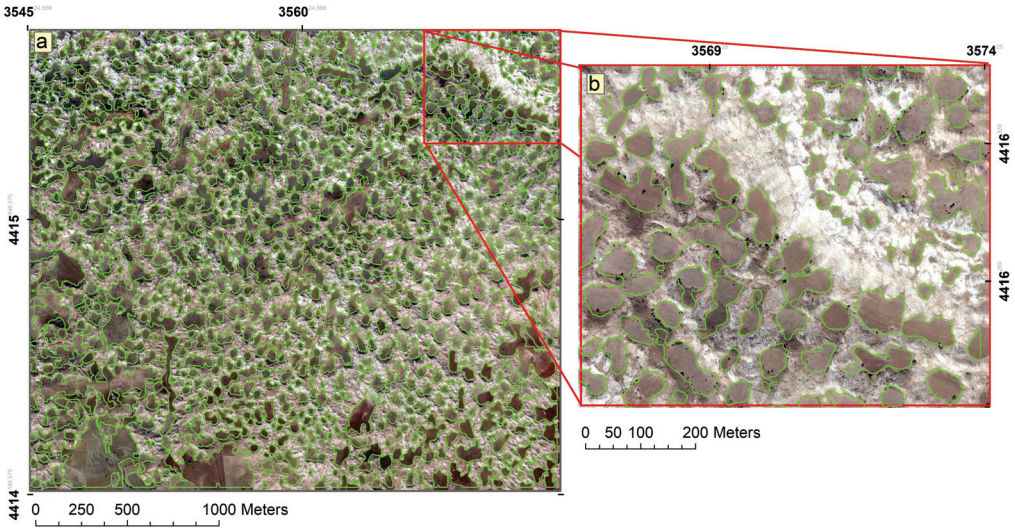


Figure 7. (**a**) Predicted dolines on test area, (**b**) Zoomed view.

In this study, the Adam optimizer with a learning rate value of 0.0001 was used as an optimization algorithm. In addition, ReduceLROnPlateau is used, which reduces the learning rate when the metric stops developing. The combination of Binary Focal Loss and Dice loss was used as the loss function. The batch size value was selected as 8. Higher batch sizes caused memory errors. The epoch value was selected as 50. Too many epochs

can cause the model to overfit, and very few epochs can cause the model to underfit. The optimal epoch value can be determined by assessing training and test results.

The image size is a factor that affects the performance of the model. All classes must be visible in the images. If the image size is small, the number of data increases, but images could be produced in which all classes are not visible. If the image size is large, the number of data decreases. Datasets of 128 × 128, 256 × 256, and 320 × 320 dimensions were created to determine the optimum size. The best dimension was determined as 320 × 320 for our data.

Image augmentation is a method to overcome overfitting trouble in deep CNNs and is widely utilized to enhance performance on a variety of applications [41]. The DenseNet121 model was tested with non-augmented data. The results show an overfitting problem and an unstable model (Figure 8). The mean IoU value of the training set is 98%, while that of the test set is 70%. In addition, the loss value is not as low as expected. Therefore, data augmentation was applied in this study and the results were improved.

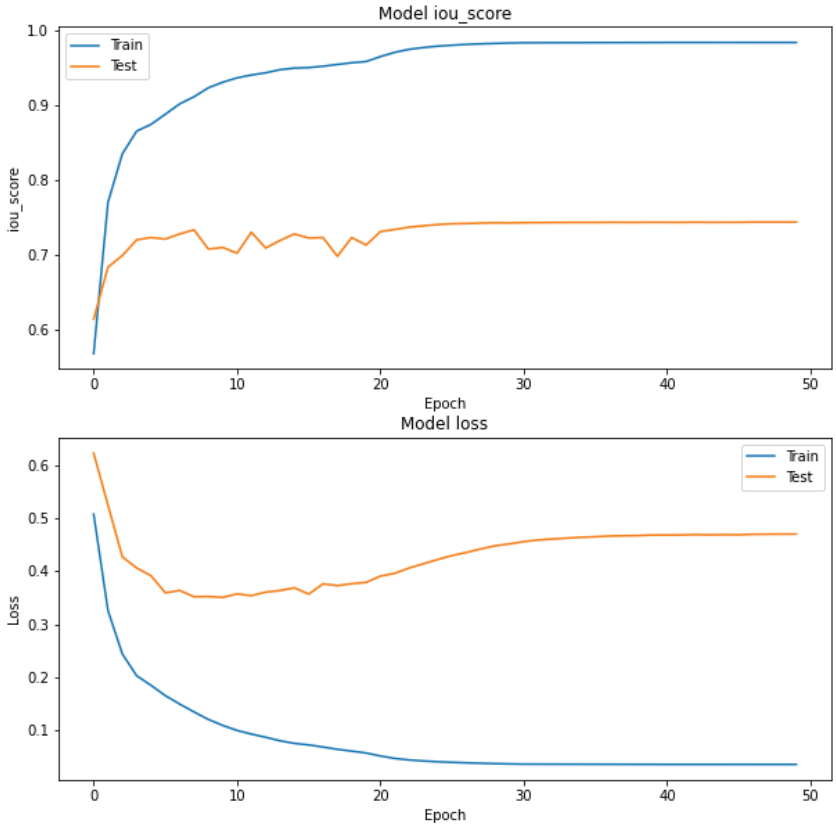


Figure 8. Model results without augmentation.

To create a robust model, it is necessary to determine the appropriate parameters. For this purpose, some parameters should be tuned. Tuning the parameters of the model requires a significant amount of time and computational resources, so researchers often decide on settings from previous experience. The settings used in similar studies can also be preferred. Also, some methods such as random search, and grid search automatically select the parameters [43,44]. In this study, loss functions, optimizers, and learning rates were tuned. Loss functions are one of the factors affecting model performance. They do not show the same performance in every model. For complex objectives such as segmentation, it is not possible to decide on a universal loss function [45]. In our study, Jaccard loss, Dice loss, and Focal Loss were tested with the DenseNet121 model. In addition, Jaccard loss–Dice loss, Binary Focal Loss–Dice loss, and Binary Cross-Entropy Loss were combined and tested. The model was tested three times for every loss function. The mean IoU values of Dice loss, Binary Cross-Entropy Loss, Jackard loss, Binary Focal Loss, and Focal Loss

were calculated as 0.7728%, 0.7755%, 0.7750%, 0.7258%, and 0.7206%, respectively. Also, some loss metrics were combined. The mean IoU values of Jaccard loss + Dice loss and Binary Focal Loss + Dice loss were calculated as 0.7494% and 0.7762%, respectively. The mean IoU value of Binary Focal Loss+Dice loss is higher than the others.

Optimizers are methods used to minimize loss function and to maximize performance. They are mathematical functions dependent on the model's weights and biases. In this study, the three most used optimizers were performed. These are Adaptive Moment Estimation (Adam), Root Mean Square Propagation (RMS-Prop), and Stochastic Gradient Descent (SGD). The learning rate is a hyperparameter that controls how much the model weights are changed with each update. This is one of the most important parameters when configuring the model. The learning rate values of 0.1, 0.01, 0.001, and 0.0001 were tested for each optimizer. The tests were performed on the DenseNet121 model, and the batch size was chosen as 8. Test IoU values were used to evaluate the results. Results show us the best optimizer is Adam with a learning rate value of 0.0001 (Figure 9). The result of the evaluation of RMS-Prop with a learning rate value of 0.0001 is very close to Adam's. The best learning rate value for the SGD optimizer was determined as 0.1.

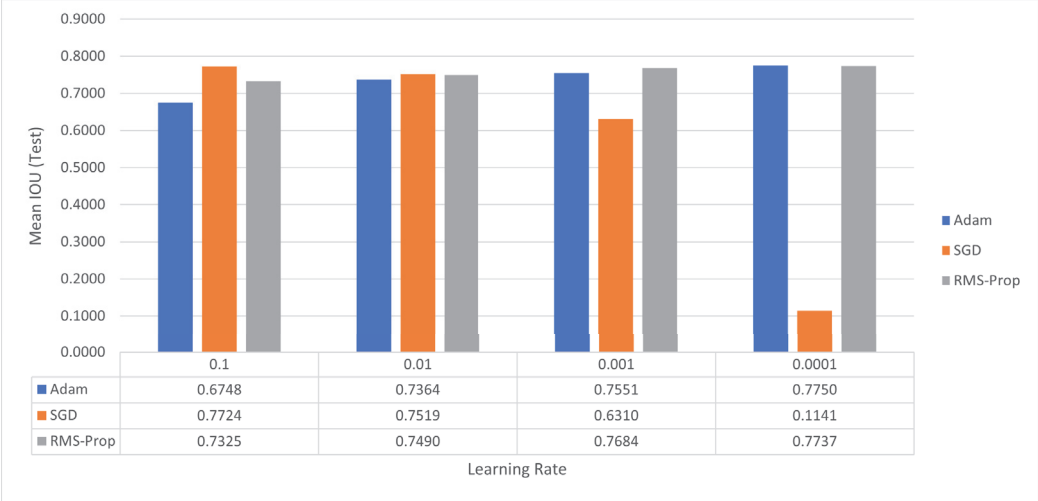


Figure 9. The performance of learning rates and optimizers.

We trained our model three times with the same dataset and the same parameters for each transfer learning method. Mean IoU and mean F-score were calculated for training and test data, and the results are shown in Table 1.

Table 1. The results of pre-trained models.

Pre-Trained Model	Train IoU	Train F-Score	Test IoU	Test F-Score
ResNet34-1	0.8433	0.9146	0.7634	0.8537
ResNet34-2	0.8533	0.9203	0.7668	0.8575
ResNet34-3	0.8386	0.9117	0.7586	0.8500
Mean	0.8451	0.9155	0.7630	0.8538
EfficientNetB3-1	0.8513	0.9191	0.7741	0.8634
EfficientNetB3-2	0.8392	0.9121	0.7728	0.8624
EfficientNetB3-3	0.8334	0.9082	0.7710	0.8598
Mean	0.8413	0.9131	0.7726	0.8619
DenseNet121-1	0.8323	0.9079	0.7712	0.8622
DenseNet121-2	0.8634	0.9265	0.7800	0.8689
DenseNet121-3	0.8490	0.9279	0.7773	0.8672
Mean	0.8482	0.9180	0.7762	0.8661

Experiments show that the results of each method are very close to each other. The best results were obtained by using DenseNet121. These are 0.8482, 0.9180, 0.7762, and

0.8661 for train IoU, train F-score, test IoU, and test F-score, respectively. These results are satisfactory for a segmentation task. IoU scores and loss graphs of the models are shown in Figure 10.

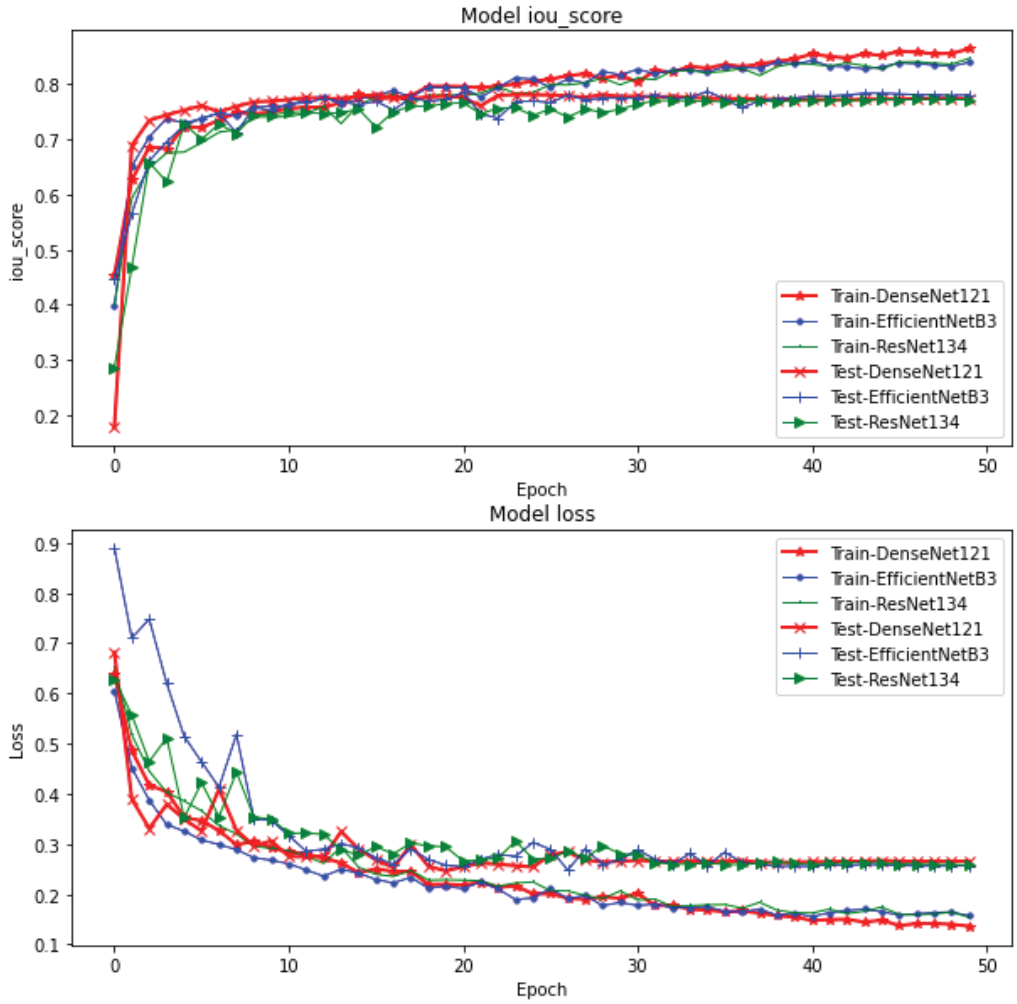


Figure 10. IoU scores and losses of models.

After the models were created, the performance of each model was examined by selecting a single image from the test dataset. For this process, images in which dolines can be observed well were selected. The IoU values of the models and the predicted masks were compared. Performances are different in each image. ResNet34 was successful on some images, DenseNet121 was successful on others, and EfficientNet B3 was successful on others. In brief, no single model is successful on all images. However, borders of predicted masks of DenseNet121 are observed more appropriately in visual control (Figure 11). These results can be combined in future study topics to benefit from all model results.

In the literature, various studies [22,23,25] have been conducted on the detection of dolines. Their common feature is that they identify areas containing dolines using geotechnical methods. Geotechnical methods are also more costly and time consuming than deep learning methods. To the best of our knowledge, there is no study in the literature that detects dolines using deep learning and creates a doline inventory map.

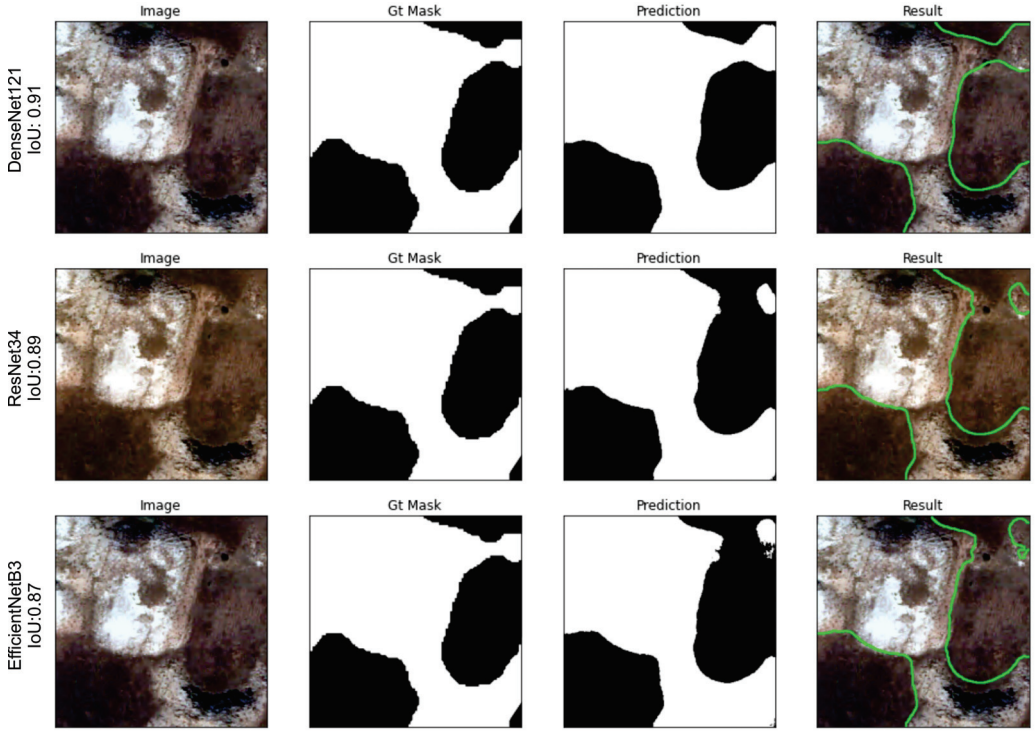


Figure 11. The performances of models for a single image.

The main objective of our article is to successfully detect and inventory uncovered collapse and dissolution dolines. To achieve this, our study utilizes the U-Net model and deep learning techniques. The proposed model gives successful results in large areas containing many dolines and where the depressions are not filled. However, distinct approaches may be necessary for areas where the dolin count is low, depressions are filled, and dolines are covered by vegetation. We recognize the need to develope specialized image processing techniques and employ appropriate suitable satellite imagery and filtering for the detection of covered dolines.

At this point, in future studies, a more specific method will be necessary for the detection and inventorying of covered dolines. This should particularly focus on which image processing techniques can be employed for the concealment and accurate detection of such dolines and which satellite images and filters should be preferred. This can be organized as part of a more comprehensive study on karst areas.

4. Conclusions

Karstic field studies are carried out to generally include measurements and evaluations made in the field. However, karstic areas can not be fully evaluated by field studies due to their characteristics and complex features. One of the reasons is their abundance in the field. Researching in these areas takes a lot of time and effort.

This study proposes a model for the fast and easy detection of dolines in the karstic areas. Deep CNN techniques were used for the segmentation of dolines. U-Net architecture, which requires less data and gives good results, is preferred for this task. All processes are completed automatically by Python scripts.

The results obtained from our model clearly reflect almost the entire karst structure in the field with all its morphological features. All predicted data were georeferenced and can be used in any GIS software. Thus, morphological measurements regarding dolines can be made easily. These results can be used in inventory studies, risk assessment studies, and geomorphological studies. Moreover, our model can be used to detect different types of landforms in future works.

The applied method introduces a novel approach to doline detection and mapping, which is not commonly found in existing methods. The results indicate that the applied method achieves a higher level of accuracy in identifying and mapping dolines, making it superior to previous techniques. Also, the method was designed to be more efficient and had allowed for faster doline detection and mapping, which can be particularly valuable for large-scale applications.

In conclusion, while this article focuses on its achievements in detecting uncovered dolines, it lays the groundwork for guiding future research. Specialized studies will be required for the detection and inventorying of covered dolines, which will be an essential step for a more comprehensive analysis of karst areas.

Based on the findings and discussions presented in the paper, the main recommendation for further research would be to expand the application of the proposed method to different geographic regions and diverse geological conditions. This would help assess the method's robustness and adaptability in various contexts and improve its generalizability for doline detection and mapping. Additionally, exploring the potential integration of other data sources, such as LiDAR or hyperspectral imagery with the existing method could enhance its accuracy and applicability. Further investigations into the scalability of the approach for larger areas and its potential for automation would also be valuable for practical applications in environmental and geological studies.

Author Contributions: Ali Polat: Conceptualization, Methodology, Software, Data curation, Formal analysis, Writing—original draft. İnan Keskin: Conceptualization, Methodology, Investigation, Writing—reviewing and editing. Özlem Polat: Conceptualization, Methodology, Investigation, Writing—reviewing and editing, Supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Computer codes related to this article can be found at GitHub: https://github.com/apolat2018/Dolin-Segmentation-UNet.git (accessed on 29 August 2023).

Acknowledgments: The authors want to thank the Prime Ministry Disaster and Emergency Management Authority for supplying orthophoto images and geological map.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alagöz, C. Türkiye'de karst olayları hakkında bir araştırma. *Türk Coğrafya Dergisi* **2014**, *1*, 86–92.
2. Thomas, B.; Roth, M. Evaluation of site characterization methods for sinkholes in Pennsylvania and New Jersey. *Eng. Geol.* **1999**, *52*, 147–152. [CrossRef]
3. Arkin, Y.; Gilat, A. Dead Sea sinkholes–an ever-developing hazard. *Environ. Geol.* **2000**, *39*, 711–722. [CrossRef]
4. Hu, R.; Yeung, M.; Lee, C.; Wang, S.; Xiang, J. Regional risk assessment of karst collapse in Tangshan, China. *Environ. Geol.* **2001**, *40*, 1377–1389.
5. Cooley, T. Geological and geotechnical context of cover collapse and subsidence in mid-continent US clay-mantled karst. *Environ. Geol.* **2002**, *42*, 469–475.
6. Doğan, U. Dipsiz Göl Kapalı Havzası'ndaki Çökme ve Sübsidans Dolinleri (Batı Toroslar). *Fırat Üniversitesi Sosyal Bilimler Dergisi* **2003**, *13*, 1–21.
7. Zhou, G.; Yan, H.; Chen, K.; Zhang, R. Spatial analysis for susceptibility of second-time karst sinkholes: A case study of Jili Village in Guangxi, China. *Comput. Geosci.* **2016**, *89*, 144–160. [CrossRef]
8. Keskin, I.; Yılmaz, I. Morphometric and geological features of karstic depressions in gypsum (Sivas, Turkey). *Environ. Earth Sci.* **2016**, *75*, 1040. [CrossRef]
9. Karimpouli, S.; Tahmasebi, P. Segmentation of digital rock images using deep convolutional autoencoder networks. *Comput. Geosci.* **2019**, *126*, 142–150. [CrossRef]
10. Rahmani, H.; Scanlan, C.; Nadeem, U.; Bennamoun, M.; Bowles, R. Automated segmentation of gravel particles from depth images of gravel-soil mixtures. *Comput. Geosci.* **2019**, *128*, 1–10. [CrossRef]
11. Chen, Z.; Liu, X.; Yang, J.; Little, E.; Zhou, Y. Deep learning-based method for SEM image segmentation in mineral characterization, an example from Duvernay Shale samples in Western Canada Sedimentary Basin. *Comput. Geosci.* **2020**, *138*, 104450. [CrossRef]
12. Kotaridis, I.; Lazaridou, M. Remote sensing image segmentation advances: A meta-analysis. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 309–322. [CrossRef]

13. Taghanaki, S.; Abhishek, K.; Cohen, J.; Cohen-Adad, J.; Hamarneh, G. Deep semantic segmentation of natural and medical images: A review. *Artif. Intell. Rev.* **2021**, *54*, 137–178. [CrossRef]
14. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
15. Bai, Y.; Mas, E.; Koshimura, S. Towards operational satellite-based damage-mapping using u-net convolutional network: A case study of 2011 tohoku earthquake-tsunami. *Remote Sens.* **2018**, *10*, 1626. [CrossRef]
16. Iglovikov, V.; Shvets, A. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv* **2018**, arXiv:1801.05746.
17. Hordiiuk, D.; Oliinyk, I.; Hnatushenko, V.; Maksymov, K. Semantic segmentation for ships detection from satellite imagery. In Proceedings of the 2019 IEEE 39th International Conference On Electronics and Nanotechnology (ELNANO), Kyiv, Ukraine, 16–18 April 2019; pp. 454–457.
18. Soares, L.; Dias, H.; Grohmann, C. Landslide Segmentation with U-Net: Evaluating Different Sampling Methods and Patch Sizes. *arXiv* **2020**, arXiv:2007.06672.
19. Pan, Z.; Xu, J.; Guo, Y.; Hu, Y.; Wang, G. Deep learning segmentation and classification for urban village using a worldview satellite image based on U-Net. *Remote Sens.* **2020**, *12*, 1574. [CrossRef]
20. Khryashchev, V.; Larionov, R. Wildfire Segmentation on Satellite Images using Deep Learning. In Proceedings of the 2020 Moscow Workshop On Electronic and Networking Technologies (MWENT), Moscow, Russia, 11–13 March 2020; pp. 1–5.
21. Marangio, P.; Christodoulou, V.; Filgueira, R.; Rogers, H.; Beggan, C. Automatic detection of Ionospheric Alfvén Resonances in magnetic spectrograms using U-net. *Comput. Geosci.* **2020**, *145*, 104598. [CrossRef]
22. Mochales, T.; Pueyo, E.L.; Casas, A.M.; Soriano, M.A. Magnetic prospection as an efficient tool for doline detection: A case study in the central Ebro Basin (northern Spain). *Geol. Soc.* **2007**, *279*, 73–84. [CrossRef]
23. Pueyo-Anchuela, O.; Casas-Sainz, A.; Soriano, M.; Pocovi-Juan, A. A geophysical survey routine for the detection of doline areas in the surroundings of Zaragoza (NE Spain). *Eng. Geol.* **2010**, *114*, 382–396. [CrossRef]
24. Nahhas, F.; Shafri, H.; Sameen, M.; Pradhan, B.; Mansor, S. Deep learning approach for building detection using lidar–orthophoto fusion. *J. Sens.* **2018**, *2018*, 7212307:1–7212307:12. [CrossRef]
25. Hussain, Y.; Uagoda, R.; Borges, W.; Prado, R.; Hamza, O.; Cárdenas-Soto, M.; Havenith, H.; Dou, J. Detection of cover collapse doline and other Epikarst features by multiple geophysical techniques, case study of Tarimba cave, Brazil. *Water* **2020**, *12*, 2835. [CrossRef]
26. Čarni, A.; Čonč, Š.; Valjavec, M. Landform-vegetation units in karstic depressions (dolines) evaluated by indicator plant species and Ellenberg indicator values. *Ecol. Indic.* **2022**, *135*, 108572. [CrossRef]
27. Pradhan, B.; Al-Najjar, H.; Sameen, M.; Tsang, I.; Alamri, A. Unseen land cover classification from high-resolution orthophotos using integration of zero-shot learning and convolutional neural networks. *Remote Sens.* **2020**, *12*, 1676. [CrossRef]
28. Abdollahi, A.; Pradhan, B.; Shukla, N.; Chakraborty, S.; Alamri, A. Multi-object segmentation in complex urban scenes from high-resolution remote sensing data. *Remote Sens.* **2021** *13*, 3710. [CrossRef]
29. Abdollahi, A.; Pradhan, B. Integrated technique of segmentation and classification methods with connected components analysis for road extraction from orthophoto images. *Expert Syst. Appl.* **2021**, *176*, 114908. [CrossRef]
30. Abdollahi, A.; Pradhan, B.; Alamri, A. SC-RoadDeepNet: A New Shape and Connectivity-preserving Road Extraction Deep Learning-based Network from Remote Sensing Data. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5617815 . [CrossRef]
31. Ghasemkhani, N.; Vayghan, S.; Abdollahi, A.; Pradhan, B.; Alamri, A. Urban development modeling using integrated fuzzy systems, ordered weighted averaging (owa), and geospatial techniques. *Sustainability* **2020**, *12*, 809. [CrossRef]
32. Abdollahi, A.; Pradhan, B. Integrating semantic edges and segmentation information for building extraction from aerial images using UNet. *Mach. Learn. Appl.* **2021**, *6*, 100194. [CrossRef]
33. Abdollahi, A.; Pradhan, B. Urban vegetation mapping from aerial imagery using Explainable AI (XAI). *Sensors* **2021**, *21*, 4738. [CrossRef]
34. Öztürk, M.; Şener, M.; Şener, M.; Şimşek, M. Structural controls on distribution of dolines on Mount Anamas (Taurus Mountains, Turkey). *Geomorphology* **2018**, *317*, 107–116. [CrossRef]
35. Yakubovskiy, P. Segmentation Models. GitHub Repository. 2019. Available online: https://github.com/qubvel/segmentation_models (accessed on 9 February 2021).
36. Chollet, F. Others Keras: Deep Learning Library for Theano and Tensorflow. 2015. Available online: https://keras.io (accessed on 12 June 2023).
37. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: https://www.tensorflow.org (accessed on 10 June 2023).
38. Poisson, A.; Guezou, J.; Ozturk, A.; Inan, S.; Temiz, H.; Gürsöy, H.; Kavak, K.S; Özden, S. Tectonic setting and evolution of the Sivas Basin, central Anatolia, Turkey. *Int. Geol. Rev.* **1996**, *38*, 838–853. [CrossRef]
39. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.

40. Polat, Ö.; Güngen, C. Classification of brain tumors from MR images using deep transfer learning. *J. Supercomput.* **2021**, *77*, 7236–7252. [CrossRef]
41. Buslaev, A.; Iglovikov, V.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A. Albumentations: Fast and Flexible Image Augmentations. *Information* **2020**, *11*, 125. [CrossRef]
42. Jaccard, P. The Distribution of the Flora in the Alpine Zone. *New Phytol.* **1912**, *11*, 37–50. [CrossRef]
43. Elgeldawi, E.; Sayed, A.; Galal, A.; Zaki, A. Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. *Informatics* **2021**, *8*, 79. [CrossRef]
44. Polat, A. An innovative, fast method for landslide susceptibility mapping using GIS-based LSAT toolbox. *Environ. Earth Sci.* **2021**, *80*, 217. [CrossRef]
45. Jadon, S. A survey of loss functions for semantic segmentation. In Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Virtual, 27–29 October 2020; pp. 1–7.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

A Deep-Learning-Based Multimodal Data Fusion Framework for Urban Region Function Recognition

Mingyang Yu [1], Haiqing Xu [1,*], Fangliang Zhou [1], Shuai Xu [1] and Hongling Yin [2]

[1] School of Surveying and Geo-Informatics, Shandong Jianzhu University, Ji'nan 250101, China; ymy@sdjzu.edu.cn (M.Y.); 2021165123@stu.sdjzu.edu.cn (F.Z.); 2021165102@stu.sdjzu.edu.cn (S.X.)
[2] School of Architecture and Urban Planning, Shandong Jianzhu University, Ji'nan 250101, China; 12949@sdjzu.edu.cn
* Correspondence: 2021160105@stu.sdjzu.edu.cn

Abstract: Accurate and efficient classification maps of urban functional zones (UFZs) are crucial to urban planning, management, and decision making. Due to the complex socioeconomic UFZ properties, it is increasingly challenging to identify urban functional zones by using remote-sensing images (RSIs) alone. Point-of-interest (POI) data and remote-sensing image data play important roles in UFZ extraction. However, many existing methods only use a single type of data or simply combine the two, failing to take full advantage of the complementary advantages between them. Therefore, we designed a deep-learning framework that integrates the above two types of data to identify urban functional areas. In the first part of the complementary feature-learning and fusion module, we use a convolutional neural network (CNN) to extract visual features and social features. Specifically, we extract visual features from RSI data, while POI data are converted into a distance heatmap tensor that is input into the CNN with gated attention mechanisms to extract social features. Then, we use a feature fusion module (FFM) with adaptive weights to fuse the two types of features. The second part is the spatial-relationship-modeling module. We designed a new spatial-relationship-learning network based on a vision transformer model with long- and short-distance attention, which can simultaneously learn the global and local spatial relationships of the urban functional zones. Finally, a feature aggregation module (FGM) utilizes the two spatial relationships efficiently. The experimental results show that the proposed model can fully extract visual features, social features, and spatial relationship features from RSIs and POIs for more accurate UFZ recognition.

Keywords: multimodal data fusion; UFZ map; spatial relationship modeling; vision transformer

1. Introduction

Today, more than half of the world's population lives in cities, yet cities cover only a tiny fraction of the Earth's surface. Asian and African countries are continuously urbanizing, and the urban population continues to grow; the world's urban population is expected to increase by 500 million by 2030 [1]. Therefore, with a growing urban population, it is critical to manage and monitor limited urban areas. An urban functional zone is a concept that describes people's different activities in a certain area, such as industrial areas, commercial areas, or residential areas [2,3]. As a basic urban unit, an accurate UFZ map is very important for urban planning, management, and decision making [4,5]. However, with the rapid development of urbanization in the world, urban functional zone maps managed by the government cannot be updated in a timely manner [6,7]. Therefore, it is crucial to make accurate and timely UFZ maps.

With the rapid development of related disciplines and technologies, high-resolution remote-sensing image data have gradually shown potential in the task of UFZ recognition. Research based on remote-sensing images continues to develop, and the use of RSIs is recognized as one of the most effective and efficient methods [8,9]. In the past, some

traditional methods, such as the scale-invariant feature transform (SIFT) [10] and the histogram of oriented gradients (HOG) [11,12], were used to identify UFZs from remote-sensing images. These methods perform well in traditional land use and land cover (LULC) classification tasks but do not perform well in urban area recognition tasks with complex structures and fine semantics. Therefore, some scholars have proposed topic models to further identify UFZs from remote-sensing imagery. Additionally, probabilistic topic models [13,14] can further exploit the potential semantic relationships of urban areas by using object features. However, such methods are still only based on low-level manual features and cannot effectively represent the complex high-level semantic feature relationships of UFZs. In recent years, computer science has developed rapidly, and the advantages of many deep-learning methods [15–17] in identifying urban functional zones and scenario-level LULC classification have become increasingly obvious. For example, benefiting from the development of the transformer [18], Wang [19] designed a U-shaped transformer network based on the UNet [20] model architecture to interpret high-resolution urban scene images and obtained satisfactory results on four challenging datasets. Zhou [21] introduced the concept of the super object (SO) and classified the urban functional area based on the methods of frequency statistics and convolutional neural networks. Du [22] mapped large-scale and fine-grained urban functional zones from remote-sensing images using a multi-scale semantic segmentation network and an object-based approach. In general, the use of high-resolution RSIs can achieve relatively good results for producing fine-grained urban area classification maps [13,23,24]. However, RSI data perform well in extracting the physical characteristics of ground objects, such as the distribution of buildings and the spatial structure of cities, but they fail to reflect the dynamically changing properties and human social activity information [25,26]. In addition, some UFZ components are visually very similar, and the method based on remote-sensing image data can extract the visual semantic features efficiently, but it cannot analyze the social semantic features between and within different functional areas. The urban functional zone contains various functional properties of regions that are directly related to human social activities [27,28]. This relationship is complex to some extent. Therefore, it is difficult to obtain high-precision UFZ classification maps using only RSI data.

It has been shown that social perception and human activities are better methods for dynamically identifying urban areas [26,29]. Socioperceptual big data that record human activity in real time are becoming increasingly available, such as points of interest (POIs) [2,30], mobile phone positioning data [31,32], social media check-in data [33], geo-tagged photos [34,35], and vehicle GPS trajectory data [36,37]. Unlike remote-sensing imagery, these data are byproducts of human social activities, and many have temporal features; therefore, they contain rich socioeconomic attributes. For example, by fusing remote-sensing image data and taxi trajectory data, Qian [38] used the road network as the basic segmentation unit to identify urban functional districts based on the residual network framework. Cao [39] proposed two strategies: enforcing cross-modal feature consistency (CMFC) and cross-modal triplet (CMT) constraints. Time-dependent social sensing signature features were extracted based on a long short-term memory network (LSTM) and a one-dimensional convolutional neural network and fused with the remote-sensing image data for more accurate urban functional area classification. POIs are the main static data in social sensing data; they are not only easy to obtain but can also provide comprehensive land use information based on human activities and geographical locations [40]. Xu [41] calculated the statistical features of POIs and fused them with remote-sensing image data to identify UFZs. Lu [42] proposed a unified deep-learning framework that can simultaneously extract visual features, social features, and spatial relationship features from RSI data and POI data. Bao [43] proposed the deeper-feature convolutional neural network (DFCNN) and integrated remote-sensing data and POI data to identify urban functional zones. There have been many efforts by scholars to recognize urban functional zones, but many of the previous studies could not effectively identify the implicit relationship between UFZs and POIs because the relationship between them is not one-to-one or

one-to-many, and the synergy mechanism between POIs and RSIs has rarely been studied. Simple fusion strategies (stacking or adding) [41,43] have difficulty taking full advantage of the complementary advantages of multimodal data. Additionally, fine-grained spatial relationships are important for identifying UFZs for the following reasons [44]. The first is that the layout of urban functional zones itself helps to reduce visual ambiguity. For example, determining whether a retail area is a commercial or residential area requires information about the surrounding area. Second, aggregating spatial information between and within different urban functional areas is conducive to determining the types of UFZs. For example, there are often recreational parks next to residential areas, while there are few large green-space areas around industrial areas. This co-occurrence relationship is very clear to some extent.

To address the above problems, we built a unified deep-learning framework, Lufz-CrossFormer (Lu-CF), which can efficiently exploit the complementary advantages between RSIs and POIs and simultaneously capture the potential spatial relationships of urban functional zones. The framework is divided into two parts: a complementary feature fusion module and a spatial-relationship-modeling module. In the first module, we use a common CNN to extract visual features from remote-sensing imagery data and a CNN with a gated attention mechanism to extract social features from point-of-interest data. To utilize the POIs conveniently, we first convert them into the corresponding hierarchical distance heatmap tensor according to the number of POI categories. Then, we use a layer-weighted model (LWM) to capture the possible co-occurrence relationship between POIs and UFZs. Finally, an adaptive feature fusion module (FFM) is used to efficiently fuse the two types of features. In the second part, the fused features are recoded according to the location relationship. We designed a new network structure based on the long-short-distance attention (LSDA) network CrossFormer [45], which can simultaneously capture the local spatial relationships and global spatial relationships of the fused features. Finally, a feature aggregation module (FGM) is used to efficiently utilize the two spatial relationships for more accurate urban functional area recognition. The main contributions of this paper are summarized as follows:

(1) We designed a unified deep-learning framework and integrated remote-sensing images and POIs to recognize urban functional zones. Our method can extract visual features, social features, and spatial relationship features from different data for more accurate urban functional zone recognition, while existing relevant studies rarely take into account all three features simultaneously.
(2) We investigated which POI categories have a greater impact on the final urban functional zone recognition accuracy, as well as the advantages of using POI data compared to using RSI data, through a series of experiments, which contributes to a further understanding of the role of multimodal data in the urban functional zone recognition task.
(3) The synergy mechanism of remote-sensing image data and point-of-interest data in the urban functional zone recognition task has rarely been studied. In this study, we used a feature fusion module to adaptively fuse the visual and social features and further analyzed the specific effects of this synergy mechanism for different urban functional areas.

2. Data and Methods

2.1. Data Source

As shown in Figure 1, we used the CSU-RSISC10 dataset [44] as the main research dataset, which was collected via Google Earth in Santa Monica, a coastal city in Los Angeles, USA, that covers an area of approximately 20 km^2. It contains 288 sample-level images with a pixel resolution of 2000 $\times$ 2000 and a spatial resolution of 0.15 m per pixel. Each sample-level image was further divided into 400 nonoverlapping patch-level images of size 100 $\times$ 100. Figure 2 illustrates the two-level hierarchical structure of this dataset. The first part of the model uses unordered patch-level images as training data. The second part

uses sample-level images containing spatial location information for training. In our experiments, we chose 70% of the data as the training set and the remaining data as the test set. For the patch-level data, the distribution of the number of different UFZ categories in the training and test sets is shown in Table 1. In addition, we downloaded 22,162 POIs from the open-source OSM website and reclassified them into 9 classes: airport, industry, supermarket, retail, hotel, institution, public service, nature, and residence. The distribution of the number of each type of POI is shown in Table 2. Finally, according to the "Code for Classification of Urban Land Use and Planning Standards of Development Land (GB 50137-2011)" and the LBCS standards, we classified the urban functional zones into commerce, industry, residence, construction, institution, transport, open space, and water.

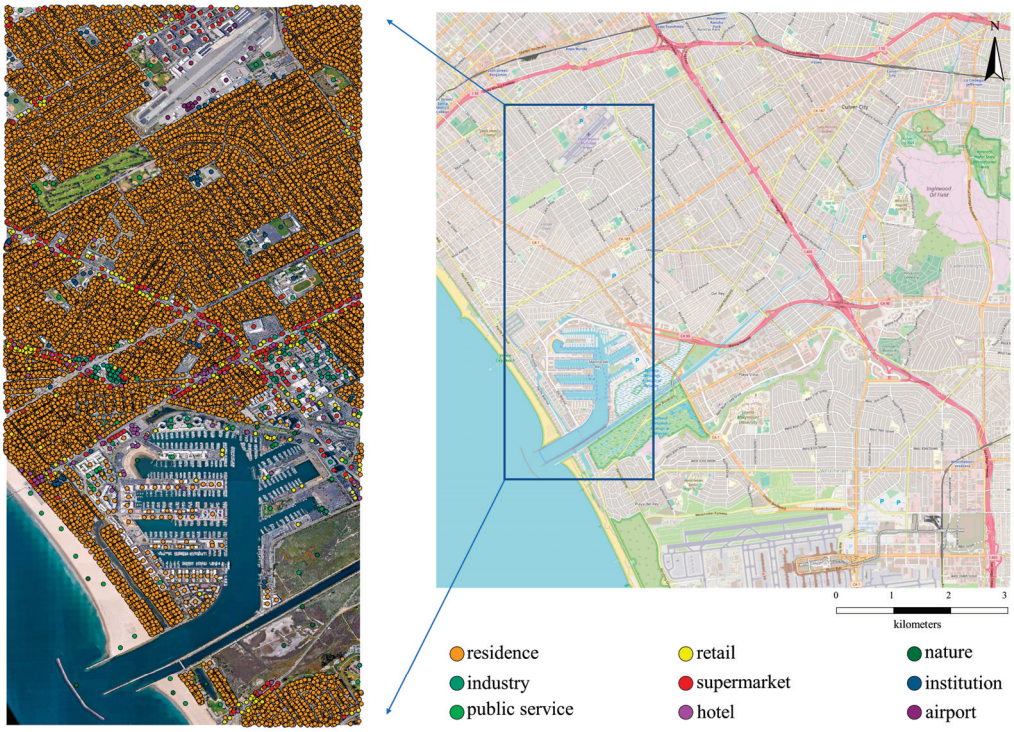


Figure 1. Overview of the dataset used in this study.

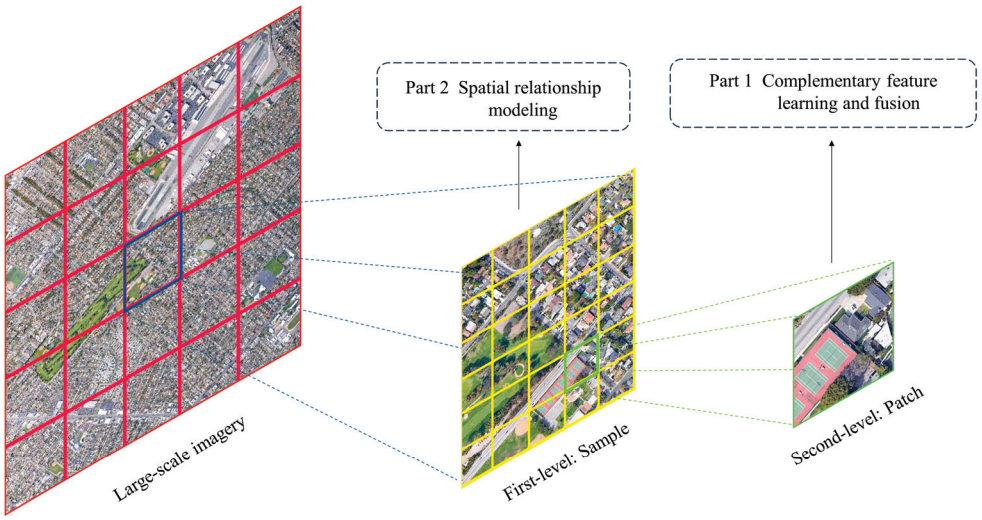


Figure 2. Two-level hierarchy of the dataset.

Table 1. Quantitative distribution of different UFZs in training and test sets.

Category	Commerce	Industry	Residence	Construction	Institution	Transport	Open Space	Water
Training set	6011	1396	30,407	346	1764	20,744	9418	10,714
Testing set	1269	698	15,127	206	570	8364	4811	3355

Table 2. Quantitative distribution of different POI categories.

Category	Residence	Retail	Supermarket	Nature	Hotel	Public Service	Institution	Industry	Airport
Number of POIs	21,045	447	310	13	91	56	62	118	20

2.2. Methods

The overall structure of the network framework established in this paper is shown in Figure 3. In the first part, we use a CNN to extract visual and social features while fusing them efficiently. In the second part, we design an architecture based on the CrossFormer network to extract local spatial relationships and global spatial relationships from the fused features generated in the first part and utilize them efficiently for the final classification output. The following subsections explain the details of the proposed framework.

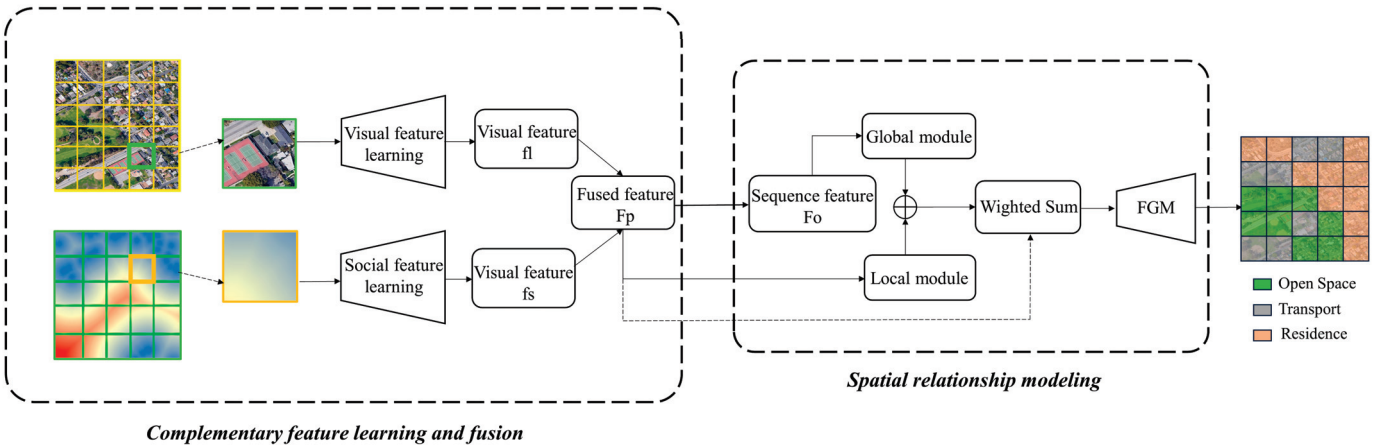


Figure 3. The overall structure of the proposed framework.

2.2.1. Complementary Feature Learning and Fusion

In this part, we use two CNNs to extract visual features and social features from RSIs and POIs. Then, the two types of features are fused efficiently to obtain fused features with strong representation ability based on the FFM.

Visual Feature Learning

First, we crop a large-scale three-band RGB RSI with a size of 3 × h × w according to the window step (h × w) to obtain a group of image units $n_j\left(j = 1, 2, 3, \ldots, \frac{H}{h} \times \frac{W}{w}\right)$, where each individual image unit represents a type of urban functional area. Then, ResNet-50 [46] is used as the backbone network to extract features from these image units, and a global average pooling operation is used to obtain the final visual feature $f_l \in \mathbb{R}^{2048\times1\times1}$.

Social Feature Learning

Many UFZs are visually similar; for example, the appearance of commercial buildings and institutional buildings may look extremely similar. Therefore, relying on remote-sensing imagery alone will not yield satisfactory results in many cases. We introduce POI data with rich social attributes as complementary features to RSI data and use a CNN with

an attention mechanism to capture possible co-occurrence relationships between urban functional zones and the corresponding points of interest.

Specifically, we first convert them into the corresponding hierarchical distance heatmap tensor according to the number of POI categories and input it into the convolutional neural network, as the CNN is better at processing two-dimensional continuous image data. Assume that the region contains M types of POIs, and the number of each POI category is n_u (u = 1, 2, …, M). For the uth-category point of interest, we calculate the minimum distance from pixel point (x_i, y_i) in the remote-sensing image to n_u POIs according to Equation (1) to generate the distance heatmap DP:

$$DP = MIN\left(\sqrt{(x_i - x_l)^2 + (y_i - y_l)^2}\right) \tag{1}$$

where (x_l, y_l) represents the spatial coordinates of the uth-category point of interest, l = 1, 2, …, n_u. As a result, we obtain M types of distance heatmaps and then stack them to form a distance heatmap tensor $\mathrm{T} \in \mathbb{R}^{\mathrm{M}\times\mathrm{H}\times\mathrm{W}}$. Similar to the remote-sensing-image-cropping method, we crop the heatmap tensor T into $H/h \times W/w$ nonoverlapping patches: $t_j \in \mathbb{R}^{M\times\mathrm{h}\times\mathrm{w}}\left(j = 1, 2, \ldots, \frac{H}{h} \times \frac{W}{w}\right)$. Second, the relationships between POIs and UFZs are not one-to-one or one-to-many. An urban functional district may have multiple types of POIs, or a POI may appear in different urban functional zones simultaneously. For example, there will be retail POIs in both commercial and residential areas. To this end, inspired by Lu's approach [42], we adaptively weight t_j by adding a layer-weighted model (LWM) to the CNN to capture possible co-occurrence relationships between different types of POIs and UFZs. The structure of the LWM model is shown in Figure 4. We first use global average pooling encoding t_j to obtain an intermediate heatmap tensor $p_j \in R^{\mathrm{M}\times 1}$, and then two fully connected (FC) layers are used to obtain the interlayer weights:

$$w_j = \varphi\left(w_2\, \delta\left(w_1 p_j\right)\right) \in \mathbb{R}^M \tag{2}$$

where $\delta(x)$ is the ReLU activation function, and $\varphi(x)$ is the sigmoid function. The first FC layer has a learnable weight $w_1 \in \mathbb{R}^{D_{r1}\times M}$ to extend the dimension of the feature t_j, and the second FC layer has a learnable weight $w_2 \in \mathbb{R}^{M\times D_{r1}}$ to reduce the output dimension of the first FC layer. Then, we use the output w_j to adaptively weight the layers of t_j by performing layerwise multiplication between t_j and w_j:

$$\left(t'_j = w_j \times t_j\right) \in \mathbb{R}^{M\times h\times w} \tag{3}$$

where t'_j represents the heatmap tensor after adaptive weighting. Finally, we incorporate the LWM into the CNN to extract the social feature $f_s \in \mathbb{R}^{2048\times 1\times 1}$.

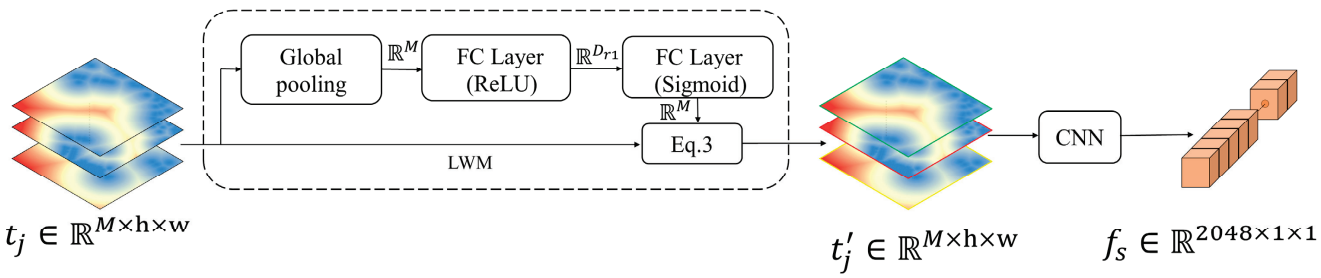


Figure 4. The structure of the layer-weighted model.

Complementary Feature Fusion

As we can imagine, visual features are effective for identifying certain urban functional areas, such as water and many open spaces. However, for commercial, industrial, and institutional areas, we need social features that contain rich socioeconomic attributes

for more accurate recognition, suggesting that visual and social features have different discriminative abilities for different UFZ categories. Therefore, similar to the Social Feature Learning section, we use an adaptive feature fusion module (FFM) for better feature fusion. As shown in Figure 5, specifically, the module can learn the adaptive fusion weights from the visual feature f_l and the social feature f_s, as shown in Equation (4):

$$w_f = \varphi(w_4\ \delta(w_3 f_c)) \tag{4}$$

where $f_c = \begin{bmatrix} f_l \\ f_s \end{bmatrix} \in \mathbb{R}^{4096}$. The first FC layer is a downsampling layer to reduce the output dimension, and its weight is $w_3 \in \mathbb{R}^{\frac{4096}{r} \times 4096}$; the second FC layer is a rescaling layer to compute the feature's weight factor, and r is the rescaling factor.

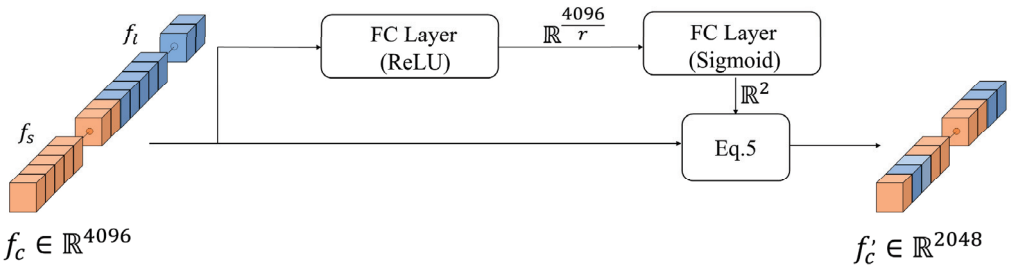


Figure 5. The structure of the adaptive feature fusion module.

$w_f = [w_l\ w_s]$ is a feature tensor of length 2, and w_l and w_s denote the ability of visual and social features to discriminate between different urban functional zones, respectively. Then, through Equation (5), we obtain the final fused feature f_c' by adaptively weighting the visual features and social features:

$$f_c{}' = w_f f_c = \left([w_l\ w_s] \begin{bmatrix} f_l \\ f_s \end{bmatrix} \right) \in \mathbb{R}^{2048} \tag{5}$$

2.2.2. Spatial Relationship Modeling

In the first part, we convert the input into a group of fused features $f_c'\left(c = 1, 2, 3, \ldots, \frac{H}{h} \times \frac{W}{w}\right)$. After that, we construct the feature tensor F_p according to the position relation of the tensor f_c', as defined in Equation (6):

$$F_p = \begin{pmatrix} f_1{}' & \cdots & f'_{\frac{W}{w}} \\ \vdots & \ddots & \vdots \\ f'_{\frac{W}{w} \times (\frac{H}{h}-1)+1} & \cdots & f'_{\frac{HW}{hw}} \end{pmatrix} \in \mathbb{R}^{2048 \times \frac{H}{h} \times \frac{W}{w}} \tag{6}$$

Each tensor f_c' in the feature tensor F_p corresponds to patch-level imagery clipping from large-scale imagery. Then, a simple convolutional network is used to extract local spatial relationships. Finally, we convert the feature F_p into a one-dimensional sequence F_o and input it into the global module to extract the global spatial relationships:

$$F_o = \left(f_1', f_2', f_3', \ \ldots\ , f'_{\frac{HW}{hw}} \right) \in \mathbb{R}^{L \times E} \tag{7}$$

where $L = \frac{H}{h} \times \frac{W}{w}$ is the sequence length, and E is the embedding dimension.

As shown in Figure 6, the local spatial relationships can be obtained through two groups of parallel convolution operations, and four groups of CrossFormer blocks (CF blocks) are used to obtain the global spatial relationships. The global features and local features are added, and the results are weighted sums with the original input F_p. Finally, a feature aggregation module (FGM) is used to obtain the final global–local context.

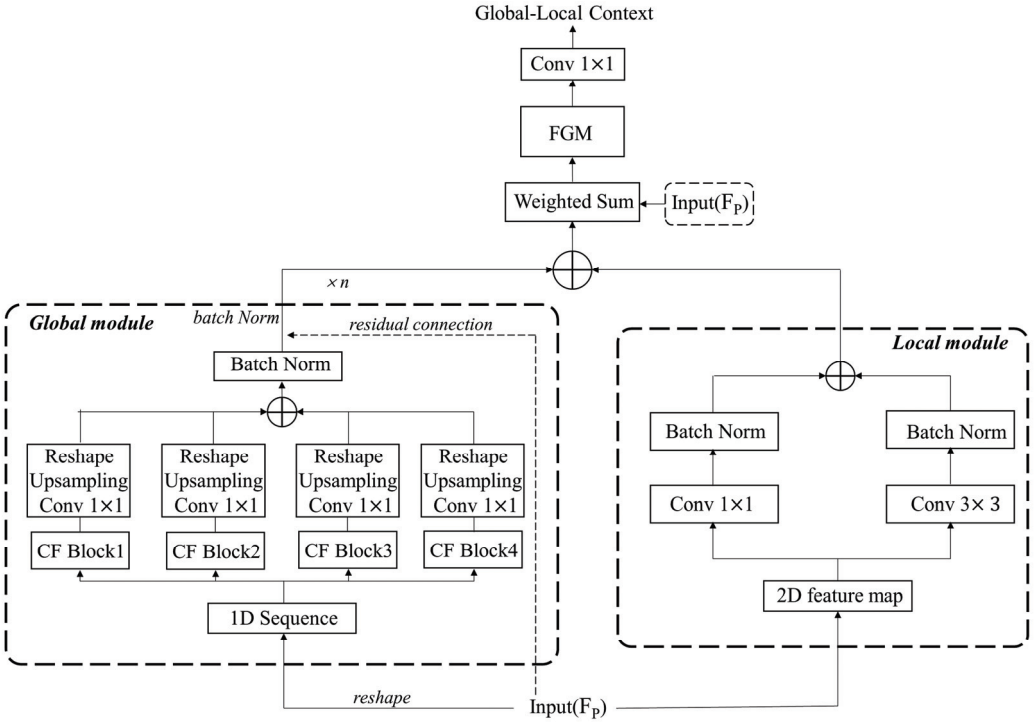


Figure 6. Illustration of the spatial-relationship-modeling module.

Local Spatial Relationship Modeling

While global spatial relationships are crucial to identifying complex UFZs, local information is also important for maintaining rich spatial details. As shown in the right half of Figure 6, we use two groups of parallel convolutions containing batch normalization operations to extract local information, with convolution kernel sizes of 1 and 3, and finally perform a sum operation.

Global Spatial Relationship Modeling

On a large scale, global information represents the relationships between different urban functional areas or a UFZ and its sub-UFZs. Some studies [19,47] have shown that vision-transformer-based structures have unique advantages in capturing global spatial relationships. In this paper, we present a network architecture designed to capture the global spatial relationships of the urban functional district based on CrossFormer. As shown in the left half of Figure 6, we first transform the original input F_p into a one-dimensional sequence and then input it into the CrossFormer blocks sequentially. As shown in Figure 7a, a CF block consists of a series of modules containing short-distance attention (SDA), long-distance attention (LDA), dynamic position bias (DPB), and multilayer perception (MLP). After that, we perform reshaping, upsampling, and convolution operations to restore the output to the size of the input. Finally, a residual connection is used to prevent model degradation, and we stack 6 global modules to obtain the final global features. In the following, we explain the details of the CrossFormer block.

(a) Dynamic Position Bias

The commonly used relative position bias (RPB) represents the relative embedding positions by adding a bias to the self-attention. The following equation represents the use of RPB to represent long-short-distance attention:

$$Attn = softmax\left(QK^T/\sqrt{d} + B\right)V \tag{8}$$

where $Q, K, V \in \mathbb{R}^{G^2 \times D}$ denote query, key, and value in the self-attention module, respectively, G is the group size, and D is the dimension of the embeddings. $\sqrt{d}$ is a constant, and $B \in \mathbb{R}^{G^2 \times G^2}$ is the RPB matrix. In the past, $B_{i,j} = B'_{\Delta x, \Delta y}$, where B' is a fixed-size matrix, and $(\Delta x, \Delta y)$ is the coordinate distance of the ith and jth embeddings. When the size of $(\Delta x, \Delta y)$ is larger than B', the size of the image or group is limited. The MLP-based module DPB is designed to solve the following problem:

$$B_{i,j} = DPB(\Delta x, \Delta y) \tag{9}$$

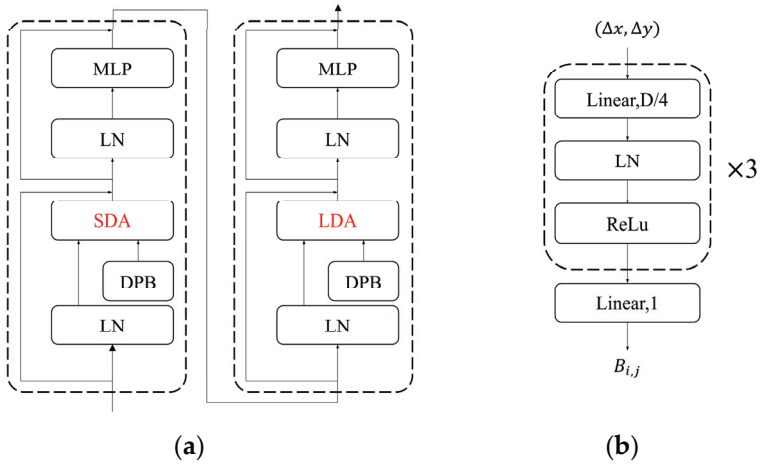


Figure 7. The structures of CF block and DPB. (**a**) CF block; (**b**) DPB.

As shown in Figure 7b, the structure comprises three FC layers containing layer normalization (LN) and ReLU activation, and the dimension of the middle layer is set to D/4. The output $B_{i,j}$ is a scalar representing the relative positions of the ith and jth embeddings. The DPB is a trainable module that can be optimized along with the entire model; it can handle arbitrary input group sizes without being limited by $(\Delta x, \Delta y)$.

(b) Long- and Short-Distance Attention

The self-attention module in the CrossFormer block is divided into two parts: short-distance attention (SDA) and long-distance attention (LDA). For SDA, we use a group of varying windows (G × G) to divide adjacent embeddings, and Figure 8a illustrates the case where G = 3. Unlike the fixed window of the Swin Transformer, SDA has a variable group size. In this experiment, the parameter G of the four CF blocks is {2, 4, 5, 4}. For LDA, given an input of M × M, the embeddings are sampled at a fixed interval I. As shown in Figure 8b (M = 9, I = 3), all of the embeddings belonging to the red boxes form a group, and the parts belonging to the purple boxes form another group. The width and height of the group can be computed by G = M/I. After grouping embeddings, both SDA and LDA compute the self-attention within their respective groups. As a result, the computational cost of the self-attention module will be reduced while capturing the fine-grained spatial relationships efficiently.

Feature Aggregation Module

The original feature F_P retains rich spatial details but lacks semantic attributes. Additionally, the global–local feature has fine-grained semantic information, but its spatial resolution is insufficient. Therefore, adding the two directly may reduce the classification accuracy [48]. In the method described in this paper, we use a feature aggregation module (FGM) to narrow the semantic gap between the two types of features to achieve more accurate UFZ recognition.

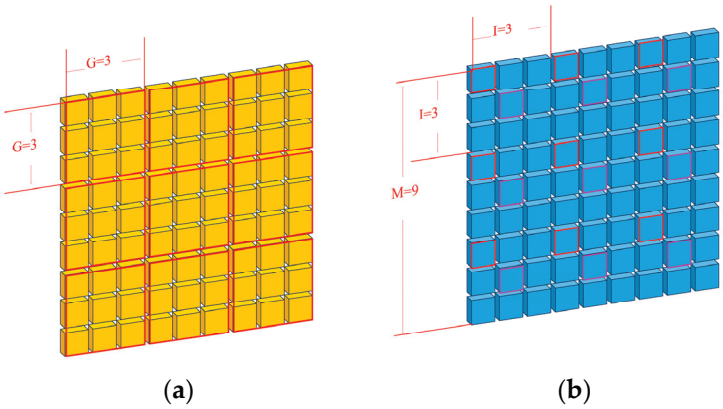


Figure 8. The structure diagrams of SDA and LDA. (**a**) SDA; (**b**) LDA.

First, a weighted-sum operation is performed on the two types of features, and the weights can be updated during model training to make full use of the rich spatial details and precise semantic attributes of different features. As shown in Figure 9, after a convolution operation (3×3), the fused features are input into the spatial path and channel path. Second, the two paths designed in the model help to strengthen the channel-based and space-based feature representation. Specifically, for the spatial path, the model uses a depth-separable convolution to produce a space-based attention feature $S \in \mathbb{R}^{h \times w \times 1}$, where h and w represent the spatial resolution of the feature map. After processing by the sigmoid function, matrix multiplication is used to obtain the path output. For the channel path, we first use the global average pooling operation to obtain the channel-based attention feature $C \in \mathbb{R}^{1 \times 1 \times c}$, where c represents the channel dimension. In addition, the rescaling operation consists of two convolution layers of 1×1, reducing and then restoring the dimension of the channel by a fixed factor. Similar to the spatial path, the sigmoid function and matrix multiplication are used to obtain the final path output. Finally, we use a convolution layer of 1×1 to obtain the FGM output.

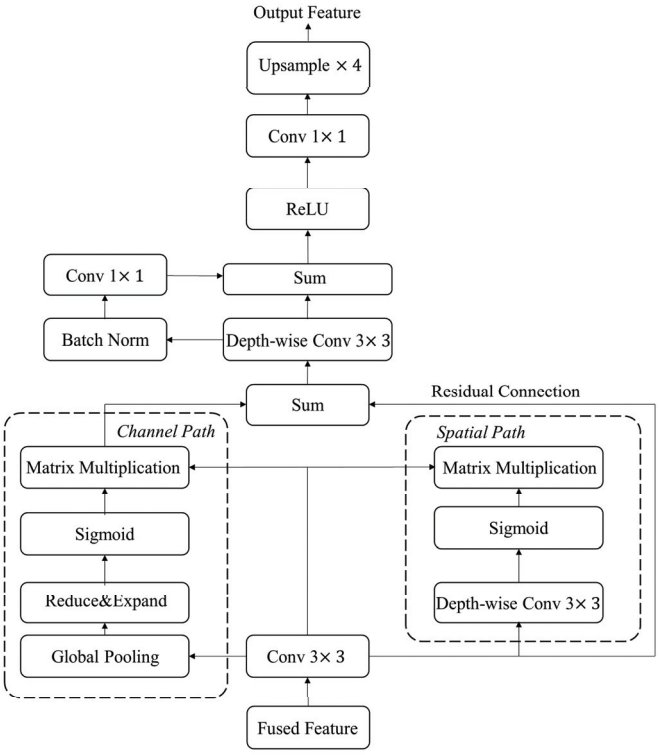


Figure 9. The structure of the FGM.

3. Experimental Analysis

3.1. Implementation Details

In the first part, we use ResNet-50 as the backbone network to extract visual features from the remote-sensing imagery. Meanwhile, the points of interest are transformed into a hierarchical distance heatmap tensor and input into the convolutional neural network to extract social features. Finally, a module based on an attention mechanism is used to efficiently fuse the two types of features. To train this part, we used cross-entropy loss (CE) and the Adam optimizer. The batch size was set to 16; the learning rate was initially set to 1×10^{-5}, and every five epochs, it became 0.98 times the original.

In the second part, we use two modules to extract global spatial relationships and local spatial relationships from the fused features, and then a feature aggregation module is used to obtain the final global–local context. After the model of the first part was trained, we saved the fused features obtained in this part to a local computer as input for the second part. Meanwhile, the loss function, optimizer, batch size, and learning rate adjustment strategies used in this part were exactly the same as in the first part.

We used the PyTorch deep-learning framework for training based on the Windows 10 operating system with an NVIDIA GeForce RTX 3090 graphics card (memory 24 GB). For the complementary feature-learning and fusion part, the model converged after 200 epochs; for the spatial-relationship-modeling part, the model converged after 150 epochs.

3.2. Evaluation Metrics

In the experiment, we used the kappa coefficient (*Kappa*) to evaluate the comprehensive performance of each model, the *F*1 score to measure the accuracy of each category, and the overall accuracy (*OA*) to evaluate the overall classification accuracy, as shown in the following equations:

$$\begin{aligned} OA &= \frac{TP+TN}{TP+TN+FP+FN} \\ precision &= \frac{TP}{TP+FP} \\ recall &= \frac{TP}{TP+FN} \\ F1 &= \frac{2\times precision\times recall}{(precision+recall)} \end{aligned} \tag{10}$$

$$\begin{aligned} p_e &= \frac{a1\times b1+a2\times b2+\cdots+aC\times bC}{n\times n} \\ Kappa &= \frac{p_o-p_e}{1-p_e} \end{aligned} \tag{11}$$

where *TP* denotes the number of pixels correctly classified into the positive class, *TN* denotes the number of pixels correctly classified into the negative class, *FP* denotes the number of pixels misclassified into the positive class, and *FN* denotes the number of pixels misclassified into the negative class. *Po* denotes the overall classification accuracy; $a1$, $a2$, …, aC denote the number of true samples in each category; $b1$, $b2$, …, bC denote the number of predicted samples in each category; and n is the total number of samples.

3.3. Experiments

3.3.1. Comparative Experiments for the Second Part

In the following experiment, we kept the structure of the first part unchanged and compared different models for the second part to further verify the importance of spatial relationships for urban functional zone recognition. The models we chose to compare include the hierarchical vision transformer networks based on the sparse attention mechanism Swin Transformer (SWINT) [49] and the pyramid vision transformer (PVT) [50], the long short-term memory network (LSTM) [51], and the gated recurrent neural network (GRU) [52] based on time-series modeling. For a fair comparison, we stack six global modules and one local module for all experimental models.

As shown in Figure 10, the area classified by our method is purer and the road structure is more complete compared with other methods. The main reason is that our model can

obtain richer composition patterns of spatial relationships by adding the variable group size and long-short-distance attention. As shown in the red circles in the first row, our method accurately identifies most of the institutional areas and has fewer misclassified spots, with a clearer road network compared to the results obtained by the GRU and LSTM. As marked by the red circles in the fourth row, our method obtains better results than SWINT and PVT due to the consideration of finer spatial relationships. In the sixth row, the classification results of our method are purer than those of the SWNT and PVT. Although the GRU and LSTM have no obvious misclassified spots in this region, the urban functional area edge recognized by them is not precise enough.

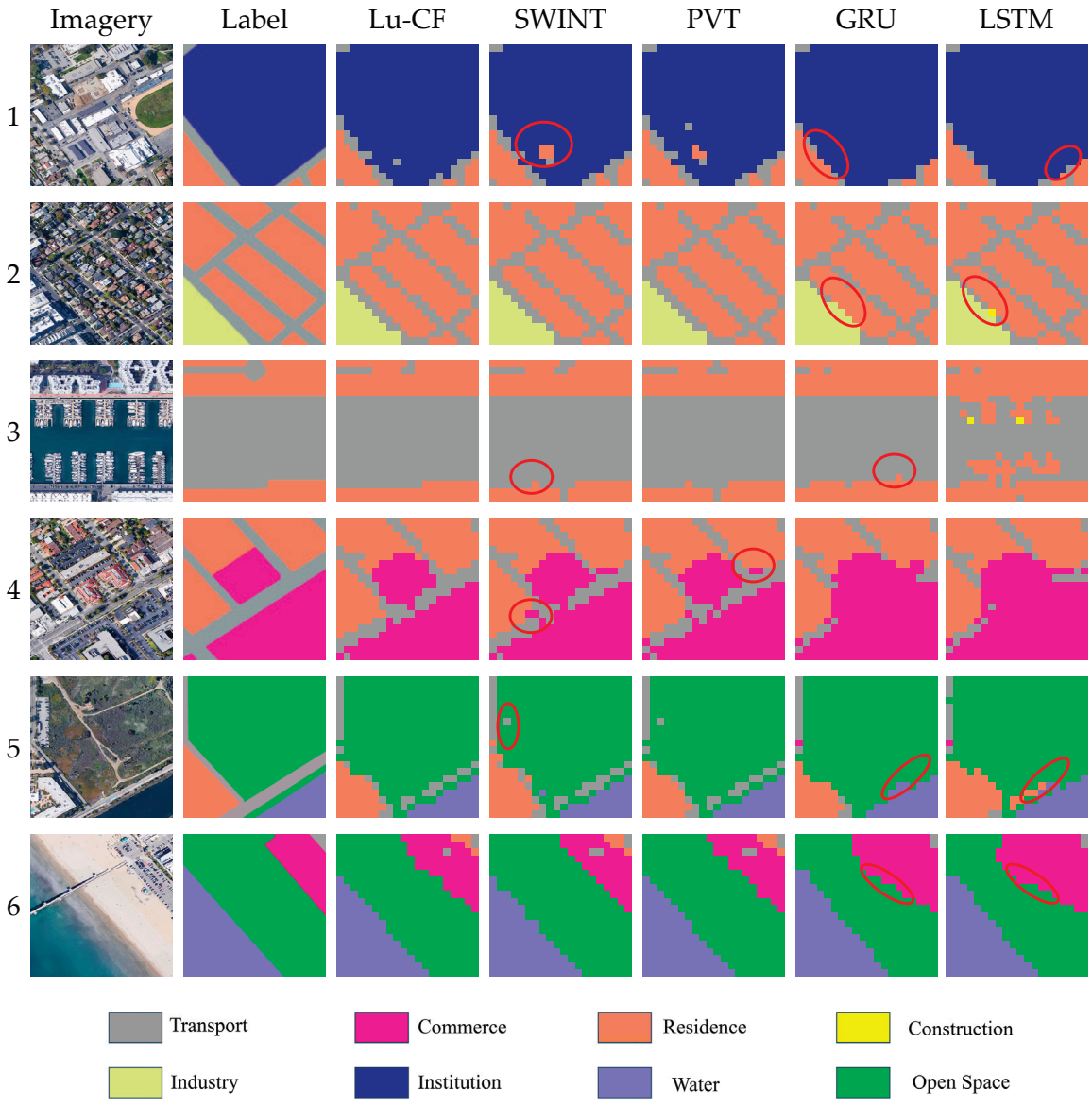


Figure 10. The test results of different models for the second part.

The quantitative evaluation results of the models are shown in Table 3. The GRU and LSTM can only capture the long-range dependence information in the horizontal and vertical directions, so they are the least effective. Our method improves *Kappa* by 1.03% over the second-best method and 7.04% over the worst method, and the overall accuracy is improved by at least 1.8% over other methods. In addition, the *F*1 scores of each UFZ category can further prove the advantage of our method. Our model has the best performance in six categories, especially the *F*1 score of the transport category,

which reaches 0.8820. Meanwhile, the accuracy performance is not satisfactory in the construction category due to the samples being scarce. For the other two categories, the accuracy gap between our method and the best method is less than 1%. The qualitative and quantitative evaluation results show that our model can make full use of the fine-grained spatial relationships for more accurate urban functional zone recognition.

Table 3. Quantitative evaluation results of different models for the second part.

Models	F1 Scores								*Kappa*	*OA*
	Transport	**Commerce**	**Residence**	**Construction**	**Industry**	**Institution**	**Open Space**	**Water**		
Lu-CF	0.8820	0.7865	0.9300	0.6528	0.8661	0.7514	0.9125	0.9596	0.8697	0.9136
SWINT	0.8796	0.7458	0.9280	0.6213	0.8669	0.7349	0.9109	0.9600	0.8594	0.8956
PVT	0.8794	0.7338	0.9276	0.6471	0.8690	0.7202	0.9105	0.9571	0.8587	0.8924
LSTM	0.7805	0.6578	0.9146	0.4512	0.8392	0.7041	0.8772	0.9474	0.7993	0.8451
GRU	0.8196	0.6707	0.9225	0.5325	0.8139	0.6938	0.8915	0.9476	0.8276	0.8762

3.3.2. Comparative Experiments with Other Methods

In this part, we used three related methods and a benchmark model for comparative experiments. The first one is the RPFM (Remote Sensing Images and Point-of-Interest Fused Model) [41], which also uses the image patch as the basic mapping unit, and it represents the relationship between POIs and urban region functions based on the distance metric. The second one is DHAO (deep integration of high-resolution imagery and OpenStreetMap) [53], which uses statistical features from POI data and visual features from RSI data for urban functional zone classification, and the road network is used as the basic mapping unit. The third one is the SO-CNN (Super-Object-based CNN) [21], which uses only remote-sensing image data to identify UFZs, and the super object is used as the basic mapping unit. In addition, the benchmark model is the widely used UNet network [20], which only utilizes image data to identify urban functional areas.

On the one hand, social perception information is crucial for accurate UFZ recognition, and our method can sufficiently capture it through the complementary feature-learning and fusion module. By learning rich social features, our approach can identify institutional areas, commercial areas, and open spaces more accurately. The results obtained by the SO-CNN are not accurate enough because only RSI data are considered. Although the RPFM and DHAO also use both POI data and RSI data, they simply incorporate the statistical characteristics of POI data and therefore fail to take full advantage of their social attributes. As a result, the inadequate representation can produce misclassified results, such as the red circles in Figure 11. We further give two specific examples. The region in the fourth row of Figure 11 consists of a large number of commercial areas as well as some residential areas. Accurately identifying commercial areas has always been a difficult problem in the UFZ classification task. Nevertheless, our method correctly identified almost all commercial areas in the region and has fewer misclassification spots. In the fifth row, our model accurately identified the area where the beach meets the ocean, while all other models misclassified this area as a transport category. On the other hand, spatial relationship features are also important for accurate UFZ classification. As marked by the yellow circles in Figure 11, the results obtained by our method are purer and more continuous due to the consideration of the spatial position relationships between different patches, while other methods only utilize the features of a single UFZ. In addition, the benchmark model obtained the worst results due to the model structure and use of a single data type.

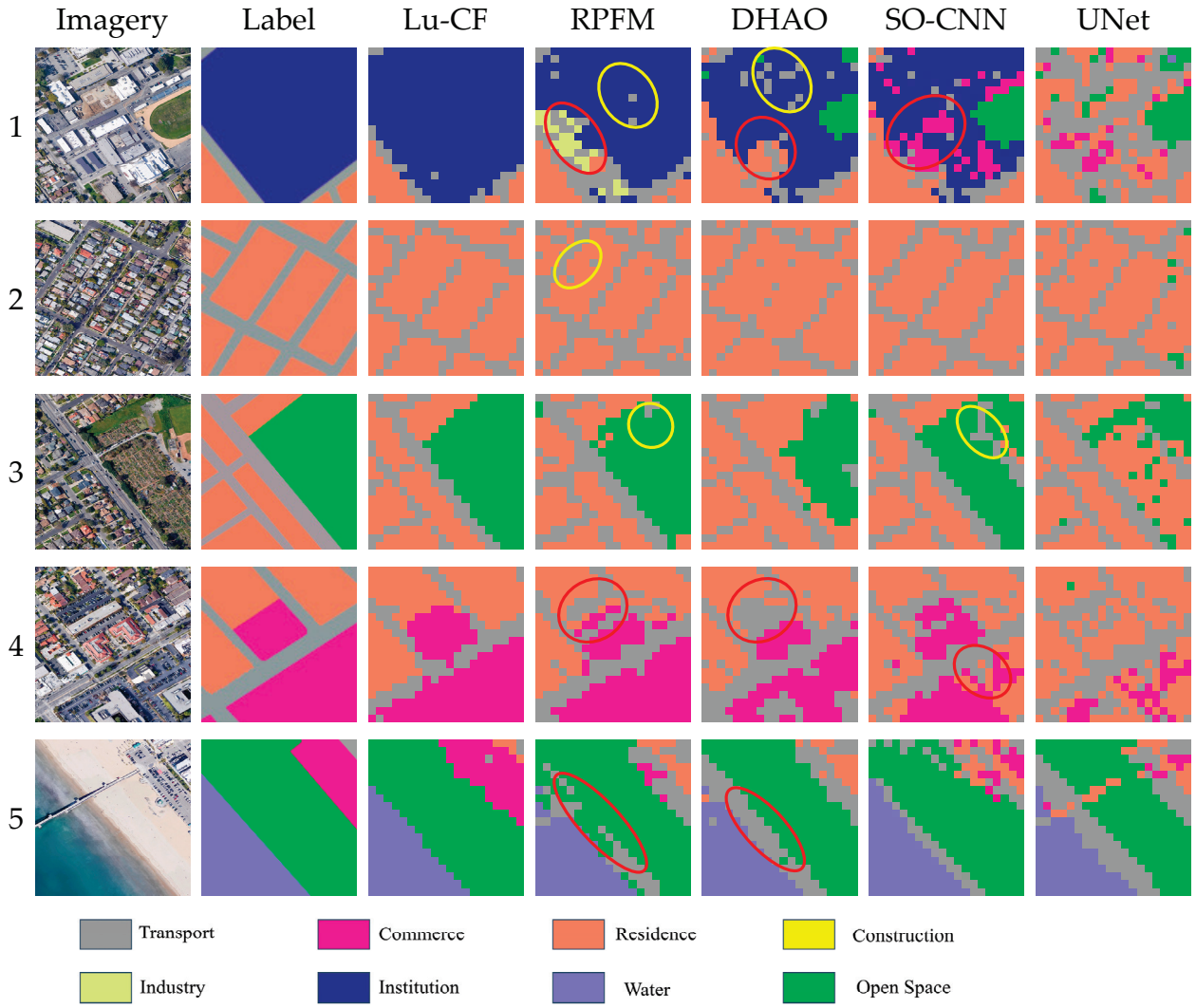


Figure 11. The test results of different models.

The quantitative evaluation results are shown in Table 4, which shows that our method obtains the highest scores on most of the metrics. The kappa value of our method is at least 2.84% higher than that of other methods, and it has an overall accuracy improvement of 3.84% over the second-best method. In addition, our model achieves the highest *F*1 scores on seven UFZ categories, with the scores improved by at least 6.50% and 3.61% for the commercial and institutional areas, respectively. The qualitative and quantitative evaluation results show that our model has a better overall performance in the urban functional zone recognition task.

Table 4. Quantitative evaluation results of different models.

Models	F1 Scores								*Kappa*	*OA*
	Transport	Commerce	Residence	Construction	Industry	Institution	Open Space	Water		
Lu-CF	0.8820	0.7865	0.9300	0.6528	0.8661	0.7514	0.9125	0.9596	0.8697	0.9136
RPFM	0.8613	0.7215	0.9063	0.6276	0.8479	0.7153	0.8995	0.9432	0.8413	0.8752
DHAO	0.8143	0.6736	0.8965	0.5928	0.8035	0.6814	0.8719	0.9361	0.8167	0.8567
SO-CNN	0.8332	0.6961	0.9325	0.6016	0.8113	0.6976	0.8817	0.9456	0.8325	0.8643
UNet	0.7516	0.6198	0.8746	0.4332	0.7652	0.5678	0.7764	0.9175	0.7361	0.7623

3.3.3. Ablation Experiment

In this section, we analyze the specific contribution of each module in the proposed framework to the final urban functional zone recognition accuracy through a series of ablation experiments. The results of the ablation experiments are shown in Table 5. We conducted experiments by continuously adding new modules to verify the contribution of different modules to the UFZ recognition accuracy. For example, using both RSI data and POI data (Stage 2) improves the *Kappa* value by 0.0793 and *OA* value by 0.0779 compared to using only remote-sensing imagery (Stage 1), and the use of the LWM module (Stage 3) increases the *Kappa* value to 0.8246; meanwhile, the addition of the FFM (Stage 4) increases the *Kappa* value from 0.8246 to 0.8379. The spatial relationship modules also have an important impact on improving the urban functional district classification accuracy. After adding the local module and global module (Stage 7), the *Kappa* value increases from 0.8379 to 0.8654. Finally, our model obtains the optimal performance (*Kappa* = 0.8697, *OA* = 0.9136) after adding the FGM (Stage 8). It is worth noting that including only the local module (Stage 6) leads to a greater accuracy improvement compared to including only the global module (Stage 5), suggesting that the local spatial relationship is also crucial to accurate UFZ classification. The results of the ablation experiments show that all of the modules of our framework have a positive effect on improving the urban functional area recognition accuracy, and using POI data leads to the greatest improvement in accuracy; therefore, we discuss the relationship between points of interest and urban functional zone recognition accuracy in the following section.

Table 5. The results of the ablation experiment. RSI: use of remote-sensing imagery. POI: use of distance heatmap tensor of POIs. LWM: layer-weighted module. FFM: feature fusion module. LOCAL: local spatial relationship modeling. Global: global spatial relationship modeling. FGM: feature aggregation module.

Stage	Strategy							*Kappa*	*OA*
	RSI	**POI**	**LWM**	**FFM**	**LOCAL**	**GLOBAL**	**FGM**		
1	√							0.7364	0.7639
2	√	√						0.8157	0.8418
3	√	√	√					0.8246	0.8579
4	√	√	√	√				0.8379	0.8713
5	√	√	√	√		√		0.8548	0.8925
6	√	√	√	√	√			0.8637	0.9012
7	√	√	√	√	√	√		0.8654	0.9058
8	√	√	√	√	√	√	√	0.8697	0.9136

4. Discussion

In this section, we discuss the contributions of different point-of-interest categories to the UFZ classification accuracy and the synergy mechanism between RSI data and POI data in the urban functional area recognition task. Finally, we visualize the layer activation of visual features.

4.1. Specific Impact of POIs on UFZ Recognition

The relationship between POIs and UFZs is not one-to-one or one-to-many and is complex in many cases. To determine which point-of-interest categories have a more important impact on the urban functional zone classification accuracy, we deleted one POI category at a time and repeated the experiment. As shown in Table 6, we find that regardless of which POI types are deleted, the final recognition accuracy decreases, indicating that all POIs have a positive effect on improving the urban functional area classification accuracy. When we remove POIs in the institution, residence, industry, and public service categories, the model accuracy decreases the most, suggesting that these POI categories have a relatively strong association with the urban functional district. In contrast, deleting retail, hotel, and

supermarket POIs has less of an impact on the classification accuracy, mainly because these points of interest can occur in almost all urban functional area categories, and there is no clear representative relationship between them and the urban functional zone.

Table 6. Impact of removing different types of POIs on model accuracy.

Drop POI Category	Institution	Residence	Industry	Nature	Airport	Public Service	Retail	Hotel	Supermarket
Kappa	0.8391	0.8345	0.8456	0.8479	0.8527	0.8443	0.8613	0.8634	0.8625
Decreasing rate of Kappa (%)	3.52	4.05	2.77	2.51	1.95	2.96	0.97	0.72	0.83
Using all POIs	*Kappa* = 0.8697								

In addition, we investigated the issue of the cost of using both remote-sensing images and points of interest simultaneously versus training with more remote-sensing imagery alone in the case of obtaining close classification accuracy. As Table 7 shows, training with both RSIs and POIs results in higher accuracy than training with only twice as much RSI data, which is close to the accuracy obtained when training with only three times as many remote-sensing images, further demonstrating the advantages of using the point-of-interest data.

Table 7. The advantages of using POI data.

RSI	POI	F1 Score								*Kappa*
		Transport	Commerce	Industry	Residence	Construction	Institution	Water	Open Space	
27,000	√	0.8339	0.6051	0.5642	0.8655	0.3156	0.3912	0.8965	0.8749	0.7257
54,000	×	0.8271	0.3925	0.2313	0.8552	0.3119	0.1756	0.9125	0.7912	0.6913
80,800	×	0.8615	0.4349	0.3723	0.8964	0.3586	0.3146	0.9324	0.8153	0.7294

4.2. Synergy Mechanism of POIs and RSIs in UFZ Recognition

The discriminative abilities of visual and social features for different urban functional zone categories are different; as we can imagine, visual features are more important for identifying water, while social features may be more crucial to identifying commercial or institutional areas. To further clarify this synergy mechanism, as shown in Figure 12, for patch-level data, we take the weights of visual feature w_l and social feature w_s to generate the point (w_l, w_s) and visualize the two-dimensional display of the discriminative ability of the different features for each UFZ category. We observe two phenomena. First, even for the same UFZ category, the fusion weights change dynamically, and the distribution of fusion weights for all urban functional areas tends to be relatively dispersed and concentrated. Second, visual features are more important for identifying urban functional zones in the transport and water categories, while social features are more crucial for recognizing the commerce, industry, institution, open space, and residence classes. However, for the residence and open space categories, although the social features are dominant, the complementary role of visual features becomes increasingly obvious. For the transport category, the rule is the opposite. For the construction area, the two types of features have a relatively equivalent status.

Based on the above discussion, we conjecture that this synergy mechanism is manifested by taking one feature as the main fusion object and the other as an auxiliary object. Nevertheless, for some urban functional area categories, when the discriminative ability of the dominant feature reaches a peak or even declines, the counterpart of the complementary feature continues to improve and approaches the dominant feature. As shown in Figure 13, we give two examples to further illustrate this synergy mechanism. In Figure 13a, it is difficult to determine whether the sports field belongs to an institutional area, residential area, or open space based on visual features alone, but the surrounding residential POIs provide

strong social signals indicating that the field belongs to a residential area; in Figure 13b, a large area of green space provides strong visual information indicating that the area belongs to the open space, while the points of interest are scarce in this area, and it will produce incorrect results if we only rely on the surrounding residential POIs. As a result, our model tends to give the social features a larger weight in Figure 13a and the visual features a larger weight in Figure 13b. Therefore, based on the different features obtained from RSIs and POIs, the synergy mechanism is a dynamic process in specific urban functional zones, and the experiments prove that the model in this paper can efficiently utilize this synergy mechanism for more accurate urban functional area recognition.

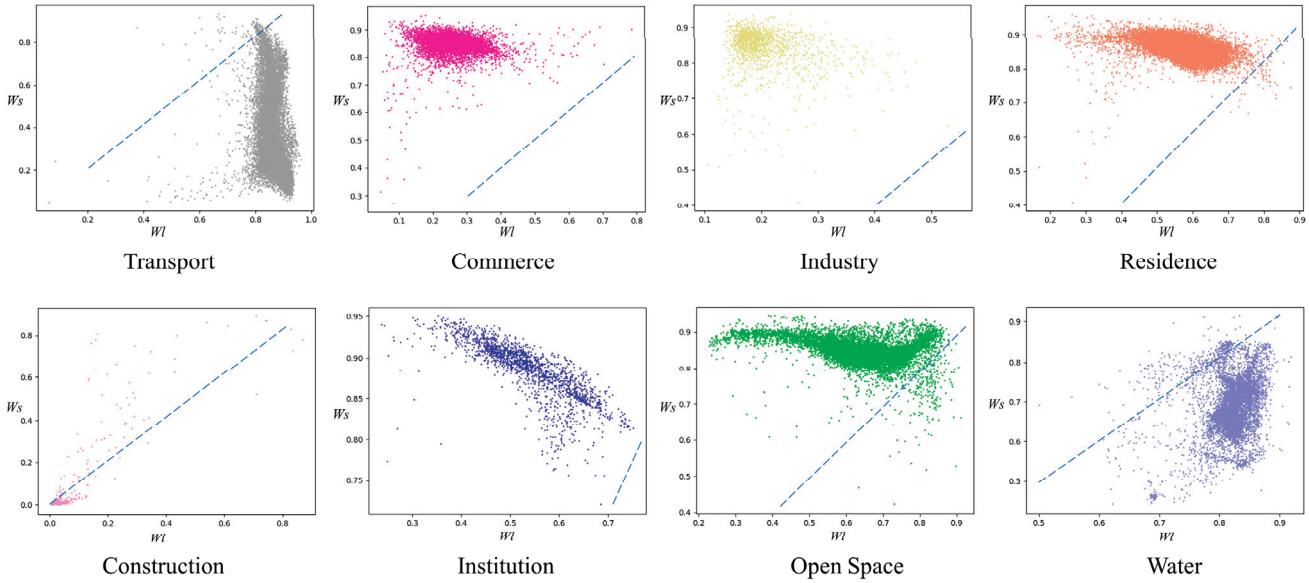


Figure 12. Two-dimensional visualization of discriminative ability of different features for different UFZ categories. *Ws* represents the weight of social features, *Wl* represents the weight of visual features, and the blue dotted line denotes the case where *Ws* = *Wl*. When the point is above the dotted line, social features are more important. When the point is below the dotted line, visual features are more important.



Figure 13. A case of the synergy mechanism. (**a**) Social features are more important; (**b**) visual features are more important.

4.3. Layer Activation Visualization of Visual Features

In the above sections, we focused on the importance of social features for urban functional zone classification; however, visual features are also crucial in almost all relevant tasks. For this purpose, we use the Grad-CAM [54] method to visualize the layer activation of the network for input during the visual feature extraction phase. As shown in Figure 14, we find that in the shallow stage (Layer 1 or Layer 2), the network tends to focus on the important regions of the image at a larger scale. When the number of layers deepens, the network focuses on a specific region of the image at a relatively small scale. Meanwhile, for urban functional districts containing more buildings, the edge and corner information of some buildings is more important; for the institutional area, we chose a relatively representative image and can see that the network mainly focuses on the areas of different surface materials on the roof; for green spaces belonging to the open space category, the network focuses on the texture information of the image; for water, the network tends to focus on the edge region or texture information of the image. As a result, visual features with strong representation ability lay a solid foundation for accurate urban functional zone recognition.

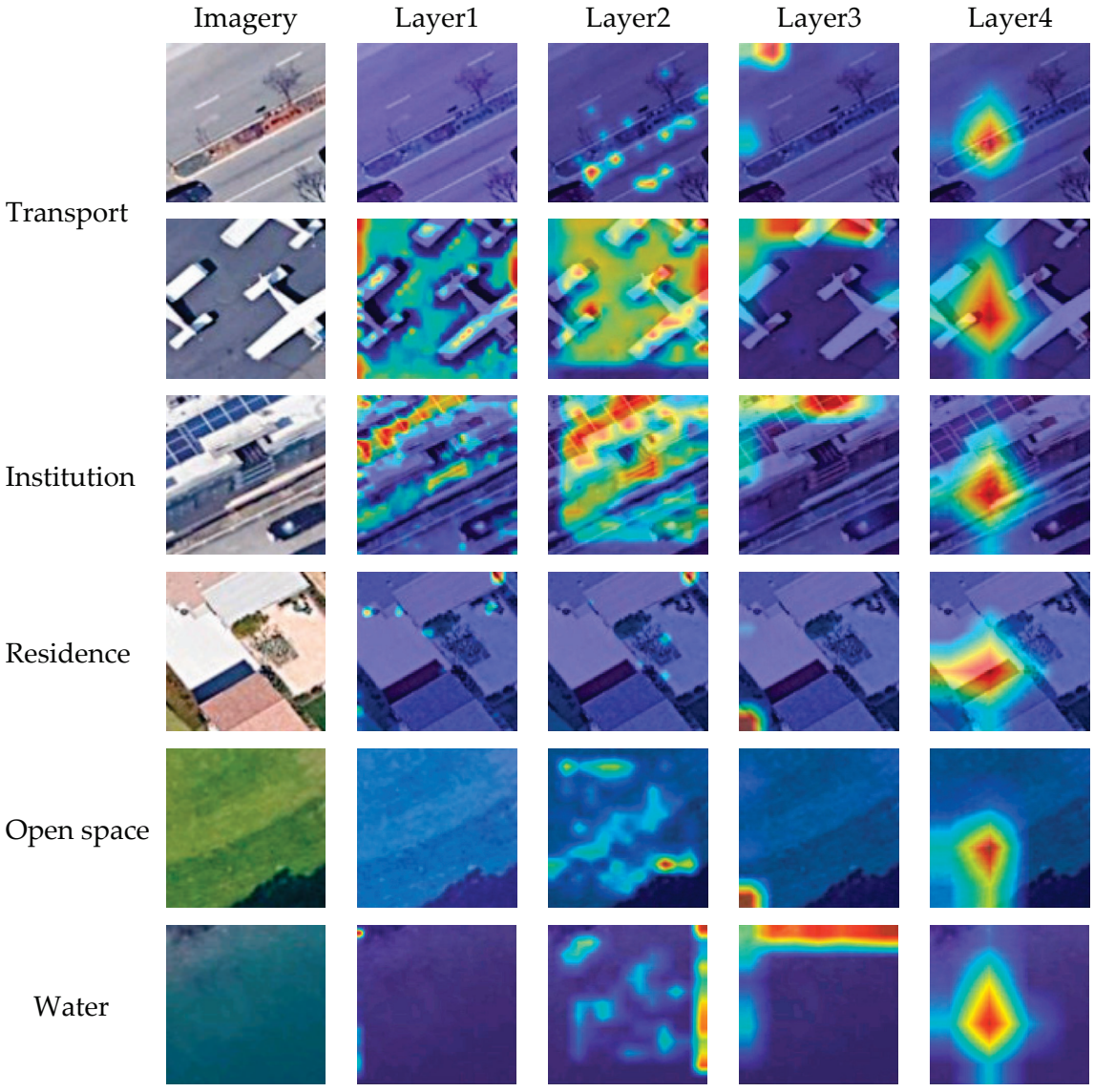


Figure 14. Layer activation visualization of visual features.

5. Conclusions

In this study, a unified deep-learning framework was designed for learning visual, social, and spatial relationship features simultaneously from remote-sensing imagery data and point-of-interest data for more accurate urban functional area recognition. In the complementary feature-learning and fusion module, we use two convolutional neural networks to extract visual features and social features from RSIs and POIs, respectively, and then the feature fusion module is used to fuse them efficiently. In the spatial-relationship-modeling module, taking the fused features obtained in the first part as input, we designed a new network structure based on the CrossFormer to extract the global and local spatial information of the urban functional zone distribution, and then a feature aggregation module is used to utilize the two spatial relationships efficiently. The comparative experimental results for the second part show that the *Kappa* value (0.8697) of our model on the test dataset is 1.03% higher than that of the second-best method (0.8594) and 7.04% higher than that of the worst method (0.7993). Additionally, it obtains the highest $F1$ scores in six urban functional zone categories, proving that the proposed model can effectively utilize the spatial relationship composition patterns of different urban functional zones to obtain more accurate results. Meanwhile, the results of comparative experiments with other methods show that the *Kappa* and *OA* values are improved by 2.84% and 3.84%, respectively, over the second-best method, proving that our method can effectively utilize the synergy mechanism based on different situations and integrates the visual features, social features, and spatial relationship features for more rational and effective urban functional area recognition tasks. In addition, we investigate the influence of different point-of-interest categories on the urban functional zone classification accuracy, the synergy mechanism of POIs and RSIs in the UFZ classification task, and the layer activation visualization of visual features. In particular, we demonstrate that the synergy mechanism is a dynamic process based on the discriminative ability of complementary features from RSIs and POIs in different UFZ categories, and our method can capture this mechanism to obtain a better urban functional zone recognition result. The framework in this paper is instructive for accurate urban functional area recognition using multimodal data. In the future, we will attempt to use more representative datasets and investigate the impact of different mapping units on the UFZ classification accuracy to achieve more efficient and accurate urban functional zone recognition tasks.

Author Contributions: Conceptualization, Haiqing Xu and Mingyang Yu; methodology, Haiqing Xu; software, Haiqing Xu; validation, Shuai Xu, Mingyang Yu and Hongling Yin; formal analysis, Haiqing Xu; investigation, Fangliang Zhou; resources, Mingyang Yu; data curation, Haiqing Xu; writing—original draft preparation, Haiqing Xu and Mingyang Yu; writing—review and editing, Mingyang Yu; visualization, Haiqing Xu; supervision, Mingyang Yu; project administration, Mingyang Yu; funding acquisition, Mingyang Yu. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China National Key R&D Program during the 13th Five-year Plan Period, grant number 2019YFD1100800, and the National Natural Science Foundation of China, grant number 41801308.

Data Availability Statement: The RSI dataset can be obtained from [44]. The POI dataset can be downloaded from https://export.hotosm.org/, accessed on 15 June 2023.

Acknowledgments: The authors thank the managing editor and anonymous reviewers for their constructive comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tu, W.; Hu, Z.; Li, L.; Cao, J.; Jiang, J.; Li, Q.; Li, Q. Portraying urban functional zones by coupling remote sensing imagery and human sensing data. *Remote Sens.* **2018**, *10*, 141. [CrossRef]
2. Hu, T.; Yang, J.; Li, X.; Gong, P. Mapping urban land use by using landsat images and open social data. *Remote Sens.* **2016**, *8*, 151. [CrossRef]

3. Zhang, Z.; Wang, Y.; Liu, Q.; Li, L.; Wang, P. A CNN based functional zone classification method for aerial images. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 5449–5452.
4. Ge, P.; He, J.; Zhang, S.; Zhang, L.; She, J. An integrated framework combining multiple human activity features for land use classification. *ISPRS Int. J. Geoinf.* **2019**, *8*, 90. [CrossRef]
5. Song, J.; Tong, X.; Wang, L.; Zhao, C.; Prishchepov, A.V. Monitoring finer-scale population density in urban functional zones: A remote sensing data fusion approach. *Landsc. Urban Plan.* **2019**, *190*, 103580. [CrossRef]
6. Yu, B.; Wang, Z.; Mu, H.; Sun, L.; Hu, F. Identification of urban functional regions based on floating car track data and POI data. *Sustainability* **2019**, *11*, 6541. [CrossRef]
7. Zhang, Y.; Li, Q.; Huang, H.; Wu, W.; Du, X.; Wang, H. The combined use of remote sensing and social sensing data in fine-grained urban land use mapping: A case study in Beijing. *Remote Sens.* **2017**, *9*, 865. [CrossRef]
8. Banzhaf, E.; Netzband, M. Monitoring urban land use changes with remote sensing techniques. In *Applied Urban Ecology: A Global Framework*; Wiley: Hoboken, NJ, USA, 2011; pp. 18–32.
9. Herold, M.; Couclelis, H.; Clarke, K.C. The role of spatial metrics in the analysis and modeling of urban landuse change. *Comput. Environ. Urban Syst.* **2005**, *29*, 369–399. [CrossRef]
10. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
11. Dalal, N.; Bill, T. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
12. Xiao, Z.; Liu, Q.; Tang, G.; Zhai, X. Elliptic Fourier transformation-based histograms of oriented gradients for rotationally invariant object detection in remote-sensing images. *Int. J. Remote Sens.* **2015**, *36*, 618–644. [CrossRef]
13. Li, M.; Stein, A.; Bijker, W.; Zhan, Q. Urban land use extraction from very high resolution remote sensing imagery using a Bayesian network. *ISPRS J. Photogramm. Remote Sens.* **2016**, *122*, 192–205. [CrossRef]
14. Zhang, X.; Du, S.; Wang, Q. Integrating bottom-up classification and top-down feedback for improving urban land-cover and functional-zone mapping. *Remote Sens. Environ.* **2018**, *212*, 231–248. [CrossRef]
15. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–16 July 2017; pp. 1251–1258.
16. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014; pp. 1–14.
17. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 2 August 2017; p. 30.
19. Wang, L.; Li, R.; Zhang, C.; Fang, S.; Duan, C.; Meng, X.; Atkinson, P.M. UNetFormer: A UNet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. *ISPRS J. Photogramm. Remote Sens.* **2022**, *190*, 196–214. [CrossRef]
20. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; pp. 234–241.
21. Zhou, W.; Ming, D.; Lv, X.; Zhou, K.; Bao, H.; Hong, Z. SO–CNN based urban functional zone fine division with VHR remote sensing image. *Remote Sens. Environ.* **2020**, *236*, 111458. [CrossRef]
22. Du, S.; Du, S.; Liu, B.; Zhang, X. Mapping large-scale and fine-grained urban functional zones from VHR images using a multi-scale semantic segmentation network and object based approach. *Remote Sens. Environ.* **2021**, *261*, 112480. [CrossRef]
23. Voltersen, M.; Berger, C.; Hese, S.; Schmullius, C. Object-based land cover mapping and comprehensive feature calculation for an automated derivation of urban structure types at block level. *Remote Sens. Environ.* **2014**, *154*, 192–201. [CrossRef]
24. Peng, F.; Weng, Q. A time series analysis of urbanization induced land use and land cover change and its impact on land surface temperature with Landsat imagery. *Remote Sens. Environ.* **2016**, *175*, 205–214.
25. Pei, T.; Sobolevsky, S.; Ratti, C.; Shaw, S.L.; Li, T.; Zhou, C. A new insight into land use classification based on aggregated mobile phone data. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 1988–2007. [CrossRef]
26. Liu, Y.; Liu, X.; Gao, S.; Gong, L.; Kang, C.; Zhi, Y.; Chi, G.; Shi, L. Social Sensing: A New Approach to Understanding Our Socioeconomic Environments. *Ann. Assoc. Am. Geogr.* **2015**, *105*, 512–530. [CrossRef]
27. Zhang, X.; Du, S.; Wang, Q. Hierarchical semantic cognition for urban functional zones with VHR satellite images and POI data. *ISPRS J. Photogramm. Remote Sens.* **2017**, *132*, 170–184. [CrossRef]
28. Zhang, C.; Sargent, I.; Pan, X.; Li, H.; Gardiner, A.; Hare, J.; Atkinson, P.M. Joint deep learning for land cover and land use classification. *Remote Sens. Environ.* **2019**, *221*, 173–187. [CrossRef]
29. Yang, H.; Wolfson, O.; Zheng, Y.; Capra, L. Urban Computing: Concepts, Methodologies, and Applications. *ACM Trans. Intell. Syst. Technol.* **2014**, *5*, 1–55. [CrossRef]
30. Liu, X.; He, J.; Yao, Y.; Zhang, J.; Liang, H.; Wang, H.; Hong, Y. Classifying urban land use by integrating remote sensing and social media data. *Int. J. Geograph. Informat. Sci.* **2017**, *31*, 1675–1696. [CrossRef]
31. Jia, Y.; Ge, Y.; Ling, F.; Guo, X.; Wang, J.; Wang, L.; Chen, Y.; Li, X. Urban land use mapping by combining remote sensing imagery and mobile phone positioning data. *Remote Sens.* **2018**, *10*, 446. [CrossRef]

32. Tu, W.; Cao, J.; Yue, Y.; Shaw, S.-L.; Zhou, M.; Wang, Z.; Chang, X.; Xu, Y.; Li, Q. Coupling mobile phone and social media data: A new approach to understanding urban functions and diurnal patterns. *Int. J. Geograph. Informat. Sci.* **2017**, *31*, 2331–2358. [CrossRef]
33. Gao, S.; Janowicz, K.; Couclelis, H. Extracting urban functional regions from points of interest and human activities on location-based social networks. *Trans. GIS* **2017**, *21*, 446–467. [CrossRef]
34. Cao, R.; Zhu, J.; Tu, W.; Li, Q.; Cao, J.; Liu, B.; Zhang, Q.; Qiu, G. Integrating aerial and street view images for urban land use classification. *Remote Sens.* **2018**, *10*, 1553. [CrossRef]
35. Zhu, Y.; Newsam, S. Land use classification using convolutional neural networks applied to ground-level images. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 3–6 November 2015; ACM: New York, NY, USA, 2015; pp. 61:1–61:4.
36. Tu, W.; Cao, R.; Yue, Y.; Zhou, B.; Li, Q.; Li, Q. Spatial variations in urban public ridership derived from GPS trajectories and smart card data. *J. Transp. Geogr.* **2018**, *69*, 45–57. [CrossRef]
37. Liu, Y.; Wang, F.; Xiao, Y.; Gao, S. Urban land uses and traffic 'source-sink areas': Evidence from GPS-enabled taxi data in Shanghai. *Landsc. Urban Plann.* **2012**, *106*, 73–87. [CrossRef]
38. Qian, Z.; Liu, X.; Tao, F.; Zhou, T. Identification of urban functional areas by coupling satellite images and taxi GPS trajectories. *Remote Sens.* **2020**, *12*, 2449. [CrossRef]
39. Cao, R.; Tu, W.; Yang, C.; Li, Q.; Liu, J.; Zhu, J.; Zhang, Q.; Li, Q.; Qiu, G. Deep learning-based remote and social sensing data fusion for urban region function recognition. *ISPRS J. Photogramm. Remote Sens.* **2020**, *163*, 82–97. [CrossRef]
40. Song, J.; Lin, T.; Li, X.; Prishchepov, A.V. Mapping Urban Functional Zones by Integrating Very High Spatial Resolution Remote Sensing Imagery and Points of Interest: A Case Study of Xiamen, China. *Remote Sens.* **2018**, *10*, 1737. [CrossRef]
41. Xu, S.; Qing, L.; Han, L.; Liu, M.; Peng, Y.; Shen, L. A New Remote Sensing Images and Point-of-Interest Fused (RPF) Model for Sensing Urban Functional Regions. *Remote Sens.* **2020**, *12*, 1032. [CrossRef]
42. Lu, W.; Tao, C.; Li, H.; Qi, J.; Li, Y. A unified deep learning framework for urban functional zone extraction based on multi-source heterogeneous data. *Remote Sens. Environ.* **2022**, *270*, 112830. [CrossRef]
43. Bao, H.; Ming, D.; Guo, Y.; Zhang, K.; Zhou, K.; Du, S. DFCNN-based semantic recognition of urban functional zones by integrating remote sensing data and POI data. *Remote Sens.* **2020**, *12*, 1088. [CrossRef]
44. Tao, C.; Lu, W.; Qi, J.; Wang, H. Spatial information considered network for scene classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 984–998. [CrossRef]
45. Wang, W.; Chen, W.; Qiu, Q.; Chen, L.; Wu, B.; Lin, B.; He, X.; Liu, W. Crossformer++: A versatile vision transformer hinging on cross-scale attention. *arXiv* **2023**, arXiv:2303.06908.
46. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
47. Zheng, F.; Lin, S.; Zhou, W.; Huang, H. A Lightweight Dual-branch Swin Transformer for Remote Sensing Scene Classification. *Remote Sens.* **2023**, *15*, 2865. [CrossRef]
48. Poudel, R.P.; Liwicki, S.; Cipolla, R. Fast-scnn: Fast semantic segmentation network. *arXiv* **2019**, arXiv:1902.04502.
49. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
50. Wang, W.; Xie, E.; Li, X.; Fan, D.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 568–578.
51. Chung, J.; Gulcehre, C.; Cho, K.H.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
52. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
53. Zhao, W.; Bo, Y.; Chen, J.; Tiede, D.; Blaschke, T.; Emery, W.J. Exploring semantic elements for urban scene recognition: Deep integration of high-resolution imagery and OpenStreetMap (OSM). *ISPRS J. Photogramm. Remote Sens.* **2019**, *151*, 237–250. [CrossRef]
54. Selvaraju, R.; Cogswell, M.; Das, A.; Vedantam, A.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

Enhancing the K-Means Algorithm through a Genetic Algorithm Based on Survey and Social Media Tourism Objectives for Tourism Path Recommendations

Mohamed A. Damos [1,2], Jun Zhu [1,*], Weilian Li [1,3], Elhadi Khalifa [2], Abubakr Hassan [2], Rashad Elhabob [4], Alaa Hm [2] and Esra Ei [2]

[1] Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu 611756, China; damosjiaotong18@gmail.com (M.A.D.); vgewilliam@my.swjtu.edu.cn (W.L.)
[2] Faculty of Engineering, Karary University, Khartoum 12304, Sudan; elhadi30@yahoo.com (E.K.); bakhas@my.swjtu.edu.cn (A.H.); alaa.hmad887@gmail.com (A.H.); esraeesa23@gmail.com (E.E.)
[3] Institute for Geodesy and Geoinformation, University of Bonn, 53115 Bonn, Germany
[4] College of Computer Science and Information Technology, Karary University, Omdurman 12304, Sudan; rashaduestc@gmail.com
* Correspondence: zhujun@swjtu.edu.cn; Tel.: +86-173-8194-4286

Abstract: Social media platforms play a vital role in determining valuable tourist objectives, which greatly aids in optimizing tourist path planning. As data classification and analysis methods have advanced, machine learning (ML) algorithms such as the k-means algorithm have emerged as powerful tools for sorting through data collected from social media platforms. However, traditional k-means algorithms have drawbacks, including challenges in determining initial seed values. This paper presents a novel approach to enhance the k-means algorithm based on survey and social media tourism data for tourism path recommendations. The main contribution of this paper is enhancing the traditional k-means algorithm by employing the genetic algorithm (GA) to determine the number of clusters (k), select the initial seeds, and recommend the best tourism path based on social media tourism data. The GA enhances the k-means algorithm by using a binary string to represent initial centers and to apply GA operators. To assess its effectiveness, we applied this approach to recommend the optimal tourism path in the Red Sea State, Sudan. The results clearly indicate the superiority of our approach, with an algorithm optimization time of 0.01 s. In contrast, traditional k-means and hierarchical cluster algorithms required 0.27 and 0.7 s, respectively.

Keywords: survey and social media tourism objectives; tourism path recommendations; genetic algorithm; machine learning algorithms

1. Introduction

The rapid proliferation of social media platforms has significantly expanded the availability of tourism objectives, enabling the recommendation of tourism paths from anywhere and at any time [1,2]. Travelers can effortlessly contribute to tourism objectives while on the go, using platforms such as WhatsApp, Facebook, Flickr, and Instagram. Additionally, users can share their current locations through services like Foursquare and provide feedback on tourist destinations via X, formerly known as Twitter. Location-based social media platforms like Foursquare and TripAdvisor empower users to share their experiences, reviews, and recommendations of tourist destinations, often including geographical coordinates [3–5]. By considering factors such as tourist travel costs, preferences, internal tourism objectives, destinations, and transportation options, it is possible to create recommended itineraries using both survey and social media tourism objectives [6,7]. The process for recommending tourist routes based on survey and social media data involves two key steps. First, the tourism objectives are clustered into groups, facilitating analysis

and recommendations. Second, the optimal tourist path is recommended by solving the traveling salesman problem (TSP) [8,9].

The increasing volume of tourism data has added complexity to data collection and clusters, particularly when suggesting optimal tourist paths using ML algorithms and solving the TSP by an optimization algorithm. These methods are instrumental in analyzing unstructured data from social media platforms, ultimately enhancing path planning and destination recommendations [10–12]. The k-means is a commonly employed clustering algorithm for tourist data. However, it faces challenges in determining the optimal number of clusters (k) and in selecting initial seeds. The choice of k significantly influences the outcomes, and researchers often employ methods like the elbow method and silhouette score for the k selection [13–15]. While effective, the elbow method may encounter difficulties when dealing with many clusters [16,17].

On the other hand, the silhouette score evaluates object similarity within clusters but may have limitations when dealing with overlapping or significantly varying-sized clusters. Selecting the initial seeds for k-means is a critical step in the process. Running the algorithm multiple times with random initialization helps identify the best outcome [18,19]. Various seed selection methods, such as random and k-means++, each come with tradeoffs involving simplicity, cluster quality, and computational efficiency [20,21]. Furthermore, the GA has emerged as a powerful optimization algorithm for addressing complex problems, including determining the appropriate number of clusters (k) and identifying optimal initial seeds for the k-means algorithm [22,23].

This paper introduces the use of the GA to address the challenges of determining k and selecting initial seeds. The GA effectively mitigates the limitations of previous methods, addressing issues such as data overlap, handling large datasets, and improving the execution time of the k-means algorithm. The GA enhances the traditional k-means algorithm by employing binary strings to represent initial centers and applying GA operators. This enhancement significantly improves the algorithm's effectiveness in classifying tourist data. Through parameter optimization, the algorithm becomes more proficient in accurately clustering and categorizing the extensive volume of tourism data, thus enhancing its overall performance. The optimization process iterates until convergence, where cluster assignments remain unchanged. Subsequently, the improved GA employs clustered and optimized tourism objectives to solve the TSP and recommend the optimal tourism path [24–27]. In summary, the main contributions of this paper are as follows:

- Enhancing the traditional k-means algorithm by using the GA to determine initial seeds, selecting the appropriate number of clusters (k), and recommending the best tourism path based on survey and social media tourism objectives.
- Collecting the tourism objectives from social media platforms through an online questionnaire and from TripAdvisor.
- Selecting and visualizing the optimal tourism path using GAs and the geographic information system (GIS) environment.
- Demonstrating the optimal time to implement the GA algorithm for finding the best tourism path through a comparison of our approach with other state-of-the-art methods.

The remainder of this article is structured as follows. In Section 2, we present the methodology and the idea of tourism objectives, survey and social media data, along with enhancing the k-means algorithm through the GA. In Section 3, the system implementation and experimental analysis are discussed. In Section 4, the results of this research and discussion are presented. Finally, the conclusion and future work are presented in Section 5.

Related Work

Tourism objectives are collected from various sources, including governmental and international institutions, geographical surveys, and popular social media platforms such as Facebook, WhatsApp, WeChat, and X. To address the static objectives of tourism planning, three main processes must be carried out: collecting tourism data, classifying tourism

data, and employing optimization algorithms to determine the best route. In the following sections, we conduct a thorough analysis of studies related to the formulation of tourism objectives, highlighting key findings and methodologies.

Hu et al. [28] presented a method for deriving tourist movement patterns from X data involving a three-step process of cleaning geo-tagged posts to identify those authored by tourists. However, this method's reliance solely on X data may limit its ability to represent comprehensive tourist activity from other sources or platforms. A separate study by Riaz and Sherani [29] overcame this limitation by focusing on the factors influencing information sharing on multiple social media platforms, particularly the adoption of Facebook and WeChat. Hashimy and T. S. [30] explored the opportunities and challenges of using social media platforms such as WhatsApp and Facebook for tourism development in Afghanistan. These include increased visibility, user-generated content, direct communication, influencer marketing, and destination marketing. However, the paper also highlights challenges that must be addressed, such as ensuring tourist safety and the need for infrastructure development. Sakas et al. [31] considered multiple objectives, including transportation type and tourist preferences, collected from various social media platforms. These objectives collectively describe the tourist destination, falling under the category of internal objectives. However, this approach overlooks the external objectives associated with interactions between tourist destinations. Addressing the challenge of integrating both internal and external objectives within a unified approach is essential for the advancement of this field.

A novel approach developed by Kim et al. [32] focused on developing a deep learning model and an image feature vector clustering technique to automate the categorization of traveler images by tourism destinations. However, the paper has limitations, primarily focusing on spatial data and omitting information about the characteristics and features of tourist destinations. The study by Bouabdallaoui et al. introduces an innovative clustering architecture that integrates the GA and k-means, coupled with a hybrid topic discovery approach incorporating latent Dirichlet allocation (LDA) and bidirectional encoder representations from transformers (BERT). The primary objective of this novel method is to predict and analyze the most significant topics related to tourist shopping destinations in Morocco. However, a significant limitation of this paper is the lack of attention in determining the values for k and the initial seeds. This limitation stems from the paper's reliance on the random selection of both the number of groups (k) and the initial seeds, raising concerns regarding the robustness and reproducibility of the results.

Yafeng et al. [33] presented a new approach based on the GA to develop 47 tourism areas in Chongqing City, China. While this paper provides an intriguing approach for applying the GA to optimize tourism path planning, which could assist tour operators and planners in developing more effective, fun, and easy tourist trips, it is essential to note that the scope of this study is centered on enhancing the planning of tourist routes specifically for the 47 scenic areas in Chongqing. As a result, the general applicability of the findings to different contexts or regions may be limited. Moreover, this research solely employed the GA to identify the optimal tourism path without delving into the diverse objectives of tourism or the various tourism data sources available, such as social media platforms. Patcharin et al. [34] introduced a method for recognizing aircraft trajectories through statistical analysis clustering. It employs k-means clustering and Gaussian mixture clustering to group unstructured trajectories observed over Suvarnabhumi International Airport. Therefore, the applicability of these findings to other regions, such as tourist destinations, may be limited. Additionally, it is worth mentioning that the k algorithm used in this approach is not optimized, which affects the algorithm's execution time. Majid et al. proposed the development of urban tourism and branding for spatial modeling. The authors used a novel hybrid modeling approach combining k-mean, fuzzy logic, and an artificial neural network (ANN) to assess urban tourism potential (AUTP). While this modeling provides valuable information for developing future strategies for urban tourism, the paper does not consider tourism data sources such as social media platforms, and it also overlooks various tourism objectives. Mehrdad et al. [35] discuss the use of unsupervised

clustering methods as data-driven models for mineral prospectivity mapping (MPM). A hybrid data-driven clustering model combines the k-means clustering algorithm with harmony search (HS) and artificial bee colony (ABC) metaheuristic optimization algorithms. This hybrid model can be used for the selection of optimum cluster centroids to highlight favorable targets in the prospecting stage of mineral explorations.

In conclusion, many papers have addressed the extraction and prediction of tourist paths based on social media platforms. However, the aforementioned papers lack a precise definition of survey and social media data in the context of tourism and its unique characteristics. They primarily analyze data from a single social media platform without comparing the suitability of various platforms for tourism research. Additionally, these papers do not introduce novel methods for analyzing social media data in the tourism domain. Furthermore, the implementation of algorithms to organize and categorize the spatial and attribute data of tourist destinations can be time-consuming.

2. Methodology

The proposed approach utilizes the combined power of the K-means algorithm and GA to optimize and cluster social media tourism data, ultimately identifying the optimal tourism path. This approach can be broken down into several stages:

First, tourism objectives are collected from social media platforms using two main methods. The first method involves distributing questionnaires on the websites of tourist groups within popular social media platforms such as WeChat and WhatsApp. This allows for the direct collection of relevant information from users. The second method involves extracting objectives from the TripAdvisor website, which serves as a valuable source of tourism objectives.

Second, once the tourism objectives have been gathered from these platforms, they are segmented into distinct groups using the K-means algorithm. The initial value of k, representing the number of clusters, is determined using the GA. Additionally, 12 specific tourism objectives are carefully selected to evaluate and assess different tourist destinations.

Third, the GA is employed to suggest and determine the best tourist path based on the clustered data. This comprehensive and systematic approach allows researchers to gain valuable insights into the preferences and patterns of tourists. These stages are all shown in Figure 1.

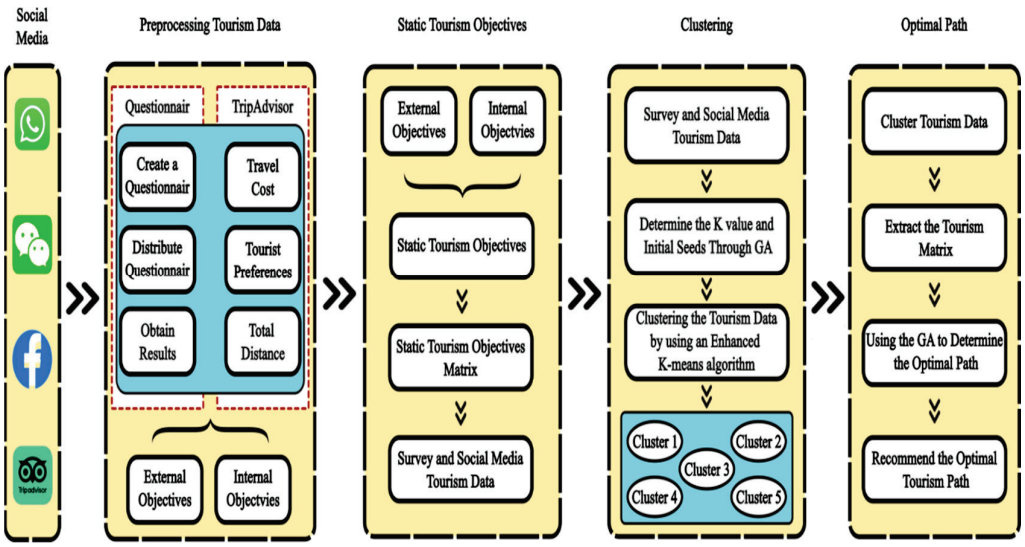


Figure 1. Process of recommending the best path based on survey and social media tourism data.

2.1. Survey and Social Media Data

Social media platforms have evolved into indispensable resources for the collection of tourism data. These platforms hold significant importance within the tourism industry, pri-

marily because they enable the real time sharing of user generated content [36,37]. Tourists utilize social media as a medium for sharing their tourism experiences, recommendations, and feedback, thus contributing to the creation of an extensive repository of information that holds immense value for researchers and tourism experts. By analyzing this rich trove of tourism data, researchers can glean valuable insights into tourist preferences. This wealth of data empowers businesses and destinations to make well informed decisions and tailor their services to align with the evolving requirements and preferences of tourists. The widespread adoption of social media platforms provides a unique opportunity to gather tourism data on a large scale, thereby fostering a deeper understanding of tourists and an overall enhancement of the tourism experience [38].

2.2. Selection Objectives

Tourism objectives in this approach were obtained from social media platforms through two methods: The first method involved the creation of a questionnaire, which was then distributed across various social media platforms groups like Facebook, WhatsApp, and WeChat. Access to the survey could be found at this link: https://forms.gle/6UHHubaiAPA6JhtE7, accessed on 25 February 2023. This questionnaire consisted of a range of inquiries concerning tourism objectives. Table 1 shows the definition of tourism objectives based on tourist preferences. Table 2 shows the sample of online questionnaire results.

Table 1. Definition of tourism objectives.

Selected Objective	Explanation and References	Evaluation Scale	Rating
Entertainment value (EN)	The value of entertainment refers to the entertainment available in the tourist site, which is available to the visitor.	Very high High Medium Low	10 7 4 1
Aesthetic and art (AA)	Aesthetics and arts include the aesthetic and artistic sensitivity. The practical, cultural, and philosophical qualities of the site.	Very high High Medium Low	10 7 4 1
Cultural–historical value (CH)	The historical and cultural value is considered to be one of the most important factors that affects why tourists flock to tourist sites.	Very high High Medium Low	10 7 4 1
Scientific value (SI)	The scientific value of the tourist site indicates the scientific importance of the site, such as universities and others.	Very high High Medium Low	10 7 4 1
Size of tourism destination (TD)	The size of the tourist destination, the height of the place, and the ability of the tourist destination to accommodate tourists.	>50 km^2 >10–50 km^2 1–10 km^2 <1 km^2	10 7 4 1
Tourism seasonality (TS)	Tourism seasonality is the possibility of visiting a tourist site in a specific season of year, some sites that can be visited year-round, such as museums, some site have seasonality such as gardens.	>300 days/year >200–300 days/year 100–200 days/year <100 days/year	10 7 4 1
Quality of service (QS)	Quality of service includes all services provided within tourist sites such as restaurants, cafes, shops, and others.	Very high High Medium Low	10 7 4 1
Time in site (TI)	This includes the time spent by the visitor inside the site, taking into account the opening and closing times of the gates.	>3 >2–3 >1–2 0–1	10 7 4 1
Biodiversity (BI)	The value of biological diversity is evaluated according to the different types of endemic animals.	Very high High Medium Low	10 7 4 1

Table 2. Sample of online questionnaire results.

Visitor NO	EN	AA	CH	SI	TD	TS	QS	TI	BI
1	low	Medium	Medium	V.high	3	3	Medium	7	low
2	Medium	low	High	V.high	4	4	V.high	5	High
3	Medium	High	V.high	High	7	5	V.high	3	V.high
4	High	V.high	Medium	V.high	5	5	Medium	10	Medium
5	Medium	low	V.high	High	3	7	V.high	7	low
6	High	Medium	low	Medium	10	5	High	5	High
7	low	low	High	Medium	7	3	Medium	7	Medium
8	low	High	Medium	V.high	3	7	V.high	5	Medium
9	High	low	High	V.high	10	4	V.high	3	V.high
10	low	V.high	low	Medium	5	7	High	7	Medium
11	V.high	Medium	low	V.high	5	10	Medium	10	V.high
12	low	Medium	High	V.high	3	4	Medium	3	V.high
13	V.high	Medium	V.high	High	7	5	V.high	5	High
14	High	V.high	Medium	V.high	5	5	low	3	Medium
15	High	V.high	Medium	V.high	10	5	High	5	High
16	low	Medium	Medium	low	10	5	V.high	7	low
17	High	V.high	Medium	V.high	5	5	V.high	5	High
18	Medium	High	V.high	High	7	5	V.high	3	V.high
19	High	V.high	Medium	V.high	3	7	V.high	7	low
20	low	low	High	Medium	7	3	Medium	7	Medium
21	High	Medium	low	Medium	3	7	V.high	5	Medium
22	low	low	High	Medium	10	4	V.high	3	V.high
23	High	Medium	low	Medium	7	3	Medium	7	Medium
24	low	low	High	Medium	3	7	V.high	5	Medium
25	High	V.high	Medium	V.high	5	5	Medium	10	Medium

The second method focused on acquiring tourist preferences, calculating distances between tourist destinations, and estimating travel costs between destinations. Table 3 proposed the tourism destination in Port Sudan City. These points were chosen on the basis of tourist demand and the aesthetic views available there, so they are considered a point of interest (POI). Tourist preferences indicate the extent to which tourists evaluate tourism destinations on the TripAdvisor website. Ratings range from 1 to 5, where 5 means very good, 4 means good, 3 means average, 2 means weak, and 1 means very weak. Table 4 displays tourist preferences according to the TripAdvisor website. Distances between tourist destinations are measured in kilometers in Table 5, and travel costs between destinations are measured in Sudanese pounds (SDG) in Table 6. This information was sourced from data collected on the TripAdvisor website.

Table 3. Tourism destinations in Port Sudan City.

POI	E	N	Name
P1	37.45045	19.72309	Sanganeb Reserve
P2	37.34297	19.11629	Othman Digna port
P3	37.33744	19.11293	Suakin city
P4	37.10517	18.76735	Arquette Resort
P5	37.10307	18.77391	Lake Arquette
P6	37.24676	19.55832	Red Sea Resort

Table 4. Matrix of tourist preferences.

POI	P1	P2	P3	P4	P5	P6
P1	0	2	4	1	3	3
P2	2	0	2	4	2	3
P3	4	3	0	1	2	3
P4	1	4	1	0	3	1
P5	3	2	2	3	0	4
P6	3	3	3	1	4	0

Table 5. Matrix of distance between tourism destinations (km).

POI	P1	P2	P3	P4	P5	P6
P1	0	67.5	68.9	112.5	111.9	29.9
P2	67.5	0	0.45	45.4	44.2	50.3
P3	68.9	0.45	0	44.11	44.4	49.9
P4	112.5	45.4	44.11	0	1.21	89.9
P5	111.9	44.2	44.4	1.21	0	88.8
P6	29.9	50.3	49.9	89.9	88.8	0

Table 6. Matrix of travel costs between destinations (SDG).

POI	P1	P2	P3	P4	P5	P6
P1	0	500	400	300	450	300
P2	500	0	250	350	300	700
P3	400	250	0	250	200	150
P4	300	350	250	0	150	400
P5	450	300	200	150	0	300
P6	300	700	150	400	300	0

2.3. Genetic Algorithm

GA is a computational method and N-hard algorithm based on the process of natural selection and genetics; it is a kind of meta-heuristic heuristic algorithm that mimics the natural process of evolution to solve complex issues [39,40]. The population of solutions in GA is randomly created and their fitness functions are evaluated to create a new population; the fittest individuals are chosen and then mutated. These operations are repeated until finding of the optimal solutions as the following:

1. Initialization: population of possible solutions is generated randomly.
2. Evaluation: each solution is tested for its fitness to use a successful fitness function.
3. Selection: more fitness functions are chosen to be the parents of the following generation.
4. Crossover: new individuals are generated by combining the existing genetic material of the selected parents.
5. Mutation: new individuals may undergo mutation, which introduces small changes in parents and a new generation replaces the old generation. The algorithm stops when a stopping parameter is satisfied, such as a set number of generations or a successful solution. Figure 2 shows the GA operations.

In general, the k-means algorithm is a machine learning tool that may be useful in a wide range of applications, such as clustering, anomaly detection, image compression, recommendation systems, and tourism path planning [41]. In our approach, we used the k-means algorithm to cluster tourism data and group it in the k group, the number of clusters was determined by using the GA.

2.4. K-Means Algorithm

The k-means algorithm is an unsupervised machine learning algorithm using clustering data, k-means classifying, and dividing data into k classes by its properties. The algorithm progresses using iterations; each data point is iteratively assigned to the closest centroid (cluster center), and the centroids are then computed again using the new assignments [42]. This procedure continues until the centroids stop moving altogether or the maximum number of iterations has been achieved. In the following are the steps of k-means algorithms:

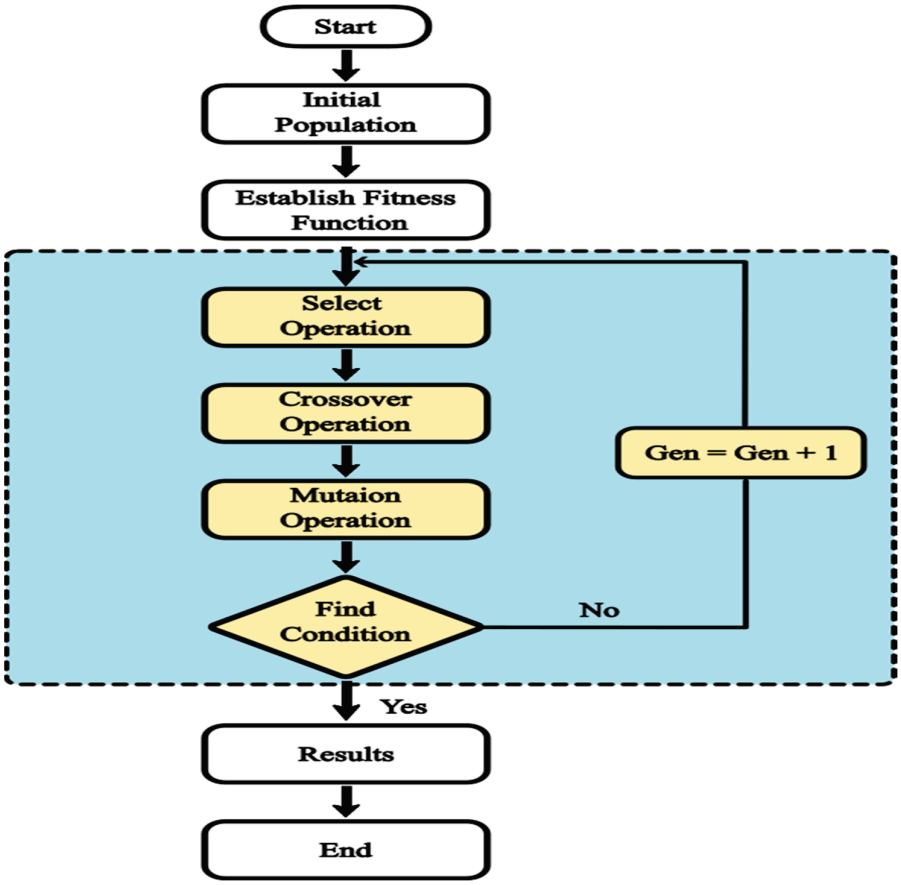


Figure 2. GA operations.

- ➢ Choose several clusters k.
- ➢ Randomly initialize k centroid.
- ➢ After the initial centroids have been selected at random, decide on each point nearest to the centroid.
- ➢ Recalculate the centroids according to the new value of the mean of all the data points in that cluster. If given two points x and y, cluster C with k data points $(x_1, x_2, \ldots, x_k)$, then the centroid C is calculated as $(\frac{1}{k}\ (x_1 + x_2 + \ldots\ldots + x_k))$.
- ➢ Repeat the 3–4 steps until the centroids stop moving altogether or the maximum number of iterations has been achieved.
- ➢ The k-means algorithm aims are determined and then find the minimized sum distance between the data and determine the centroid. Many methods can be used to determine distance such as the Euclidean distance method; this method is most commonly used if given two points x and y and the Euclidean distance is calculated as Equation (1):

$$D_{x,y} = \sqrt{(X_2 - X_1)^2 - (Y_2 - Y_1)^2} \quad (1)$$

where n is the number of data points.

2.5. Enhancing the K-Means Algorithm though GA

This approach enhances k-means to utilize GA to determine the optimal initial seeds and number of the k value for k-means clustering. Figure 2 provides an overview of the entire framework for GA k-means clustering and enhancing the algorithm showing in pseudo code. Presented below is a detailed explanation of enhancing the k-means clustering process:

In the first step of this approach, a value is defined to generate the initial population in GA and solution fitness is assessed based on clustering quality, measured by the sum of squared errors (SSE), which is utilized to determine the optimal initial seeds and the value of k. To enhance the performance of GA, optimization parameters were adjusted, as shown in Tables 3–6 these parameter adjustments aim to optimize the GA for more effective initial seed selection and k value determination for clustering. The improved GA, designed to enhance the K-means algorithm, is presented in the pseudo code.

In the second step of this process, once the optimal values for both the number of clusters (k) and the initial seeds have been determined, the tourism data are partitioned into groups. This partitioning is accomplished using the improved k-means algorithm introduced in the first step. The enhanced k-means algorithm effectively assigns each data point to its corresponding cluster based on similarity, taking into account the optimized k value and the initial seeds. By partitioning the tourism data into groups, this step facilitates further analysis and enables the identification of unique patterns, visitor preferences, or notable characteristics within the dataset.

In the third step, after segmenting the tourism data into groups using the enhanced k-means algorithm and considering the identified tourism objectives, an improved GA is employed to recommend the optimal tourist path. Although harnessing the capabilities of the GA, which combines elements of natural selection and genetic operators, the algorithm efficiently searches for the most favorable path that aligns with the specified tourism objectives. Figure 3 and Algorithm 1 shows the Framework of recommended tourism path enhancing the k-means algorithm through GA.

Algorithm 1: Pseudo code of enhancing k-means by GA

Begin
Initialization
Generate a solution population, representing possible data clusterings, cluster centers in k-means.
Fitness Evaluation
Assess solution fitness based on clustering quality, often measured by sum of squared errors (SSE)
Selection
Choose parent solutions for the next generation, with higher fitness solutions having a better chance.
Crossover
Create new solutions by combining features from two parents averaging cluster centers.
Mutation
Randomly alter some new solution features to maintain diversity and prevent premature convergence.
Replacement
Replace some current solutions with the new ones.
Termination
If a stopping criterion is met, stop and return the best intial seeds and k value found. Otherwise.
Repeat
From step 2.
Print:
Else
Print: Fail
end if
end

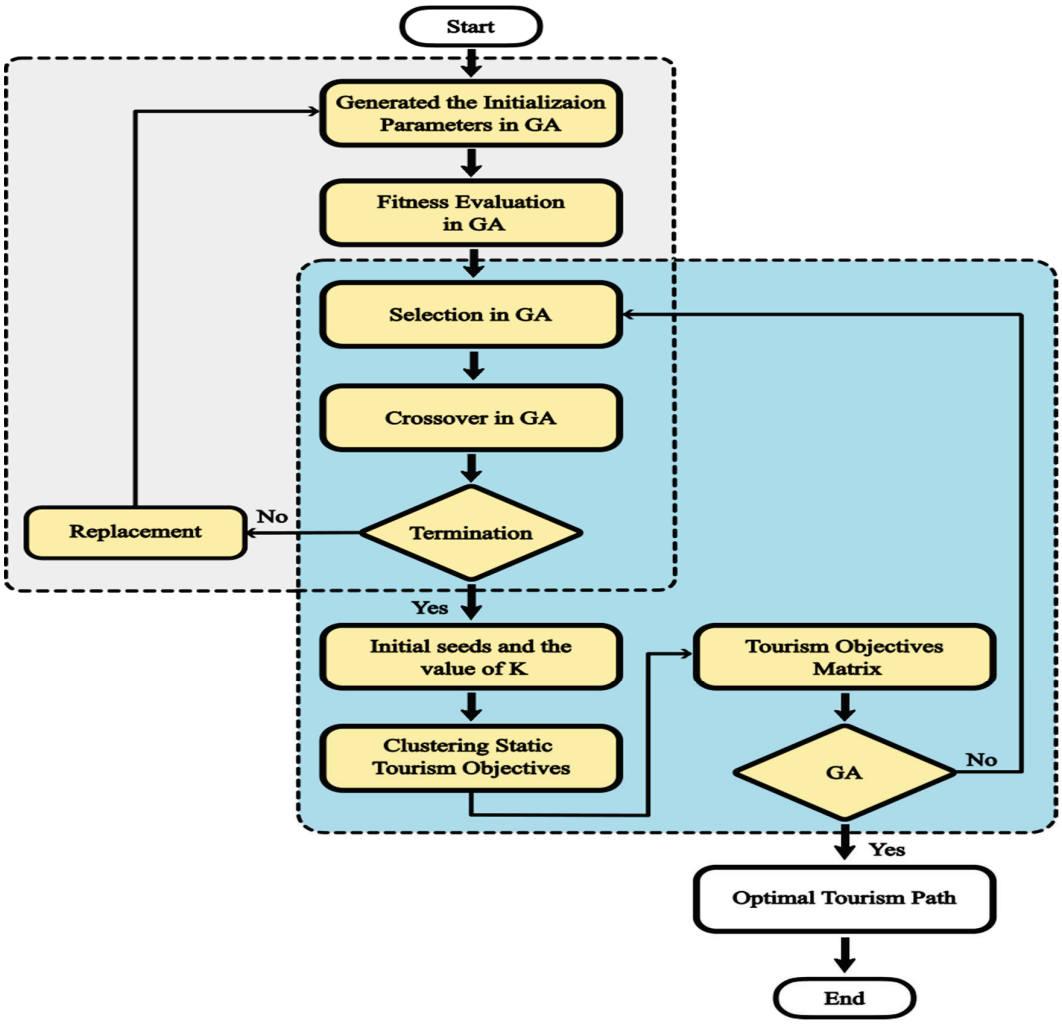


Figure 3. Framework of recommended tourism path enhancing the k-means algorithm through GA [40].

3. System Implementation and Experimental Analysis

Port Sudan City is in Red Sea State, Sudan. Port Sudan, the capital of Red Sea State in eastern Sudan, functions as Sudan's primary seaport. Over 90% of Sudan's international trade flows through Port Sudan's modern port facilities, which were built between 1905 and 1909 to replace the historical Arab port of Suakin. Port Sudan features key infrastructure like an international airport, an oil refinery, and state of the art cargo and passenger terminals. Located on the eastern coast of the Red Sea, the port handles significant volumes of container traffic, bulk commodities, and roll-on/roll-off shipments. Port Sudan serves as a strategic gateway for the landlocked countries of South Sudan, Ethiopia, and Eritrea. It provides access to key trade routes like the Suez Canal and the Bab el Mandeb Strait. With ample developable land and a deep-water harbor, the port has potential for significant expansion to support Sudan's growing trade volumes and improve supply chain connectivity. However, challenges remain around port efficiency and infrastructure constraints that have hindered full realization of Port Sudan's potential as a regional shipping hub. Ongoing developments and investments aim to address these issues, upgrade port facilities, digitize processes, and expand container handling and logistics services. If successful, such efforts could transform Port Sudan into a modern logistics center that enhances Sudan's trade competitiveness and links the nation to global supply networks. Figure 4 shows Port Sudan City in Red Sea State Sudan. Sex tourist destinations distributed in the study area

were selected. Table 3 and Figure 5 proposed this point of interest (POI) in Port Sudan city. Following the selection, questionnaires were distributed to these chosen destinations.

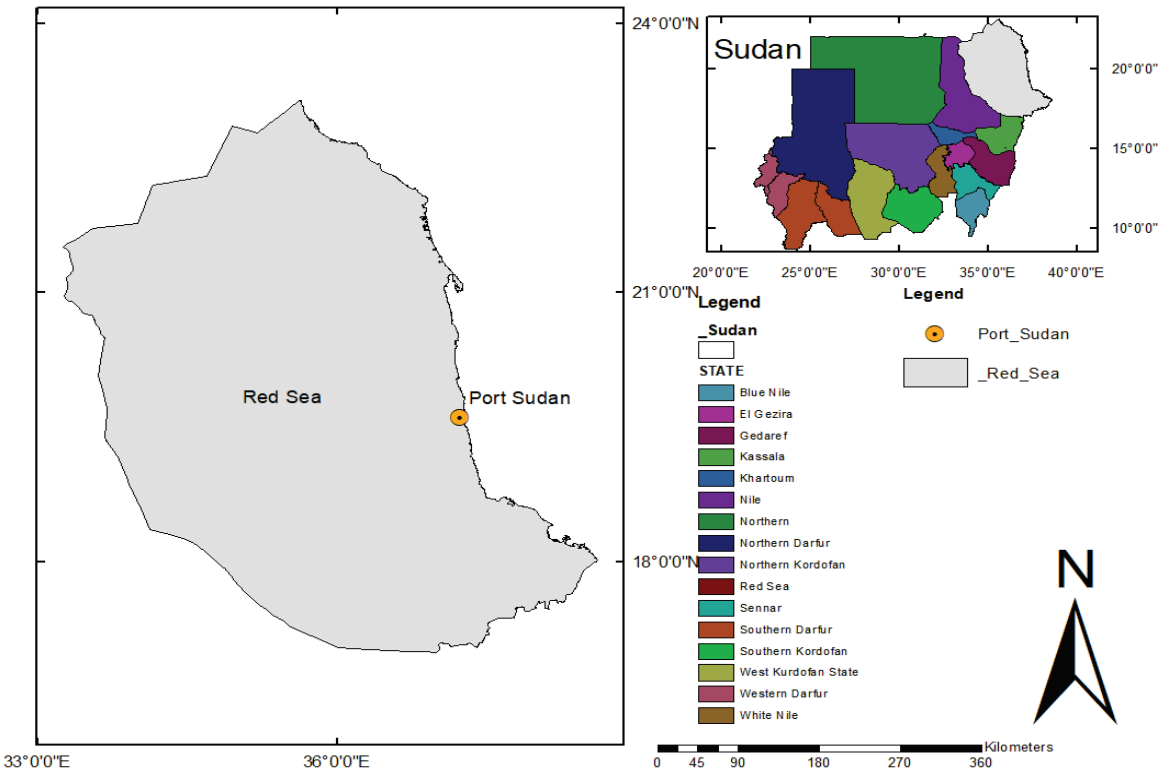


Figure 4. Location of Red Sea State Sudan in Sudan.

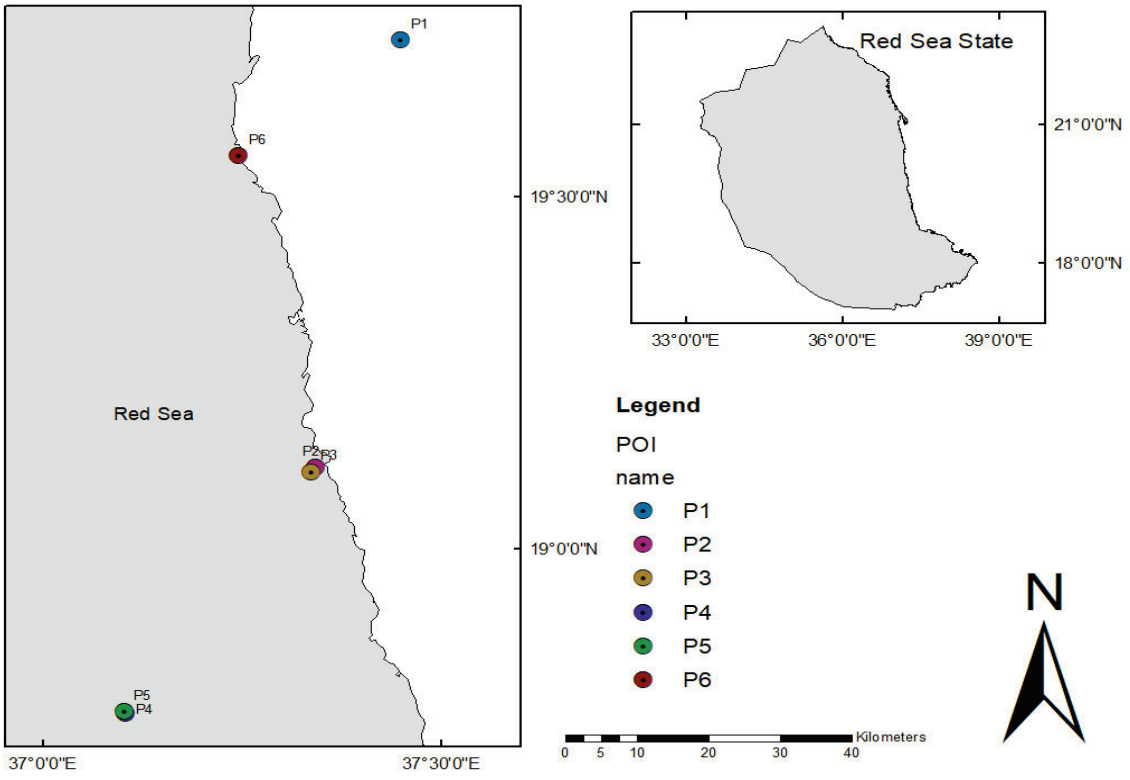


Figure 5. Tourist sites distributed within the city.

4. Results and Discussion

4.1. Results

Based on the results of online questionnaires distributed on social media platforms in the study area, we determined the static tourism objectives with input from 600 visitors to enhance our approach. We collected the external tourism objectives from the TripAdvisor social media website. To implement our approach, first, we improved the GA by using new parameters, and then we determined the optimal k value and initial seeds. The optimal k value is 5, and the optimal initial seed selection is 10. Second, we used the enhanced k-means algorithm to cluster the static tourism objectives based on the value of k. Five groups of visitors were created for each tourism objective. The optimal path recommendations for visitor destinations, determined through the enhanced GA with improved parameters in Table 7, are presented in Table 8.

Table 7. Parameter settings of GA.

Parameters	Values
Population size	100
Crossover probability	0.85
Mutation probability	0.10
Number of generations	4000

Table 8. Optimal tourism paths.

NO	The Objectives	The Path
1	EN	P1-P5-P6-P3-P2-P4-P1
2	AA	P4-P2-P1-P4-P6-P5-P4
3	CH	P5-P2-P4-P6-P3-P1-P5
4	SI	P3-P4-P6-P1-P2-P5-P3
5	TD	P2-P4-P6-P3-P1-P5-P2
6	TS	P4-P6-P4-P1-P2-P3-P4
7	QS	P6-P1-P5-P2-P3-P4-P6
8	TI	P6-P3-P2-P4-P5-P1-P6
9	BI	P5-P3-P4-P1-P2-P6-P5
10	Tourist preferences	P1-P5-P6-P3-P2-P4-P1
11	Travel costs	P6-P5-P2-P1-P3-P4-P5
12	Total distances	P2-P3-P4-P5-P6-P1-P2

Table 9 and Figure 6 show groups of the internal tourism objectives based on enhancing the k-means algorithm.

Table 9. Results of internal tourism objective groups of visitors using the enhance k-means algorithm.

Group 1						
POI/Objective	P1	P2	P3	P4	P5	P6
EN	138	113	101	113	122	131
AA	125	150	95	187	102	151
CH	122	120	135	121	78	88
SI	135	125	131	99	90	135
TD	80	89	106	87	91	78
TS	74	123	105	173	85	190
QS	78	111	135	109	106	136
TI	92	109	105	135	120	198
BI	145	189	143	143	153	110

Table 9. *Cont.*

Group 2						
POI/Objective	P1	P2	P3	P4	P5	P6
EN	102	123	132	124	164	122
AA	168	108	132	120	160	199
CH	139	121	144	108	130	151
SI	109	125	154	90	170	176
TD	175	131	131	124	119	190
TS	123	187	89	138	118	120
QS	95	108	156	149	67	99
TI	84	125	139	120	77	98
BI	101	106	76	132	90	121
Group 3						
POI/Objective	P1	P2	P3	P4	P5	P6
EN	134	153	90	129	93	106
AA	110	109	77	109	113	77
CH	167	137	87	111	156	127
SI	140	171	100	109	128	118
TD	160	126	178	160	143	98
TS	155	103	90	131	210	76
QS	123	161	124	129	186	67
TI	93	150	159	140	89	127
BI	90	135	87	98	77	145
Group 4						
POI/Objective	P1	P2	P3	P4	P5	P6
EN	111	86	137	97	107	109
AA	97	86	98	96	107	86
CH	99	148	155	167	116	150
SI	104	90	114	119	98	115
TD	104	110	82	143	163	109
TS	107	117	98	77	109	62
QS	113	150	93	87	111	120
TI	190	160	107	120	109	121
BI	99	78	130	118	160	129
Group 5						
POI/Objective	P1	P2	P3	P4	P5	P6
EN	115	125	140	137	114	132
AA	100	147	198	88	118	87
CH	73	74	79	93	120	84
SI	112	89	101	183	114	56
TD	81	84	103	86	84	125
TS	141	70	218	81	78	152
QS	191	70	92	126	130	178
TI	141	56	90	85	205	56
BI	165	92	164	109	120	95

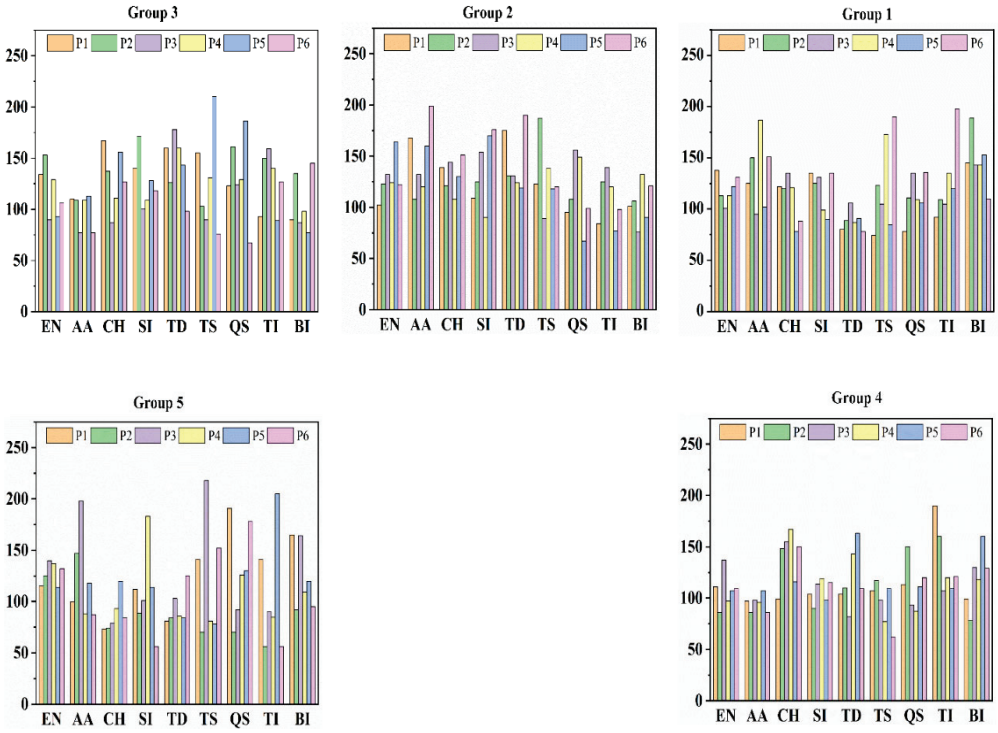


Figure 6. Groups of the internal tourism objectives based on enhancing the k-means algorithm.

After creating tables categorizing tourism objectives into groups, we constructed a comprehensive tourism objectives matrix comprising 12 matrices, nine for internal objectives and three for external objectives. The numbers in the matrices represent the numerical differences between the numbers of visitors in the EN groups across various destinations. For instance, the value 25 in the EN objectives matrix corresponds to the difference in the number of visitors between destination P1 and destination P2 in Group 1 in Table 9. Consequently, we calculated the matrices for all nine internal objectives in the same manner as illustrated in Table 10. Following this, we calculated 12 optimal paths based on static tourism objectives using the improved GA. Our approach involved creating objective matrices to address the TSP.

Table 10. EN objective matrix.

POI	P1	P2	P3	P4	P5	P6
P1	0	25	37	25	16	7
P2	25	0	55	37	48	1
P3	37	55	0	14	57	47
P4	25	37	14	0	9	36
P5	16	48	57	9	0	13
P6	7	1	47	36	13	0

4.2. Discussion

When collecting and classifying tourism data, online surveys and social media play significant roles. In our approach, we propose to enhance the k-means algorithm for optimizing tourism data obtained from social media platforms. The primary challenge lies in determining the optimal k value for the k-means algorithm and selecting initial seeds, which can be addressed using various methods. Our method is based on the fundamental premise of plotting different cost values against varying k values. The elbow point on the graph can be used to compute k, representing the point of diminishing returns or the inflection point at the elbow [43,44]. However, the drawback of the elbow method is that it occasionally struggles to produce effective clusters. As an alternative, we

employ an improved GA in our approach to determine the ideal value of k. Compared to other methods, the two-stage GA represents a relatively recent development. Many scholars have employed the k-means algorithm to classify and organize data into groups because of advantages such as ease of implementation, scalability to handle large datasets, guaranteed convergence, the ability to initialize centroids, and adaptability to new data points. However, one challenge associated with the k-means algorithm is the estimation of the optimal number of groups [41,45]. The k-means algorithm is favored by many scholars for data classification due to its strengths, such as ease of implementation and scalability. One of its drawbacks is the difficulty in determining the number of k groups. Our approach employs the GA to overcome this limitation.

In the process of clustering with k-means, initial seeds for clustering are selected. The method used to choose these seeds is dependent on the data and the problem being addressed. One approach commonly employed involves the random selection of initial seed points; the algorithm is executed multiple times, retaining the seeds that yield the lowest clustering error. Alternatively, an initial seed selection algorithm can be utilized, which selects the initial seeds from different clusters within the dataset. The decision regarding these methods is dictated by the characteristics of the data and the clustering objectives at hand [46]. In this approach, we utilize the GA to determine the optimal k value and to select initial seeds. Comparisons were conducted with several algorithms commonly used in the field of clustering. These include the expectation-maximization (EM) algorithm, hierarchical clustering, and the traditional k-means algorithm. Table 11 presents the comparison between the enhancing k-means algorithm and other machine learning clustering algorithms. If we wish to implement machine learning algorithms, the most valuable parameter is the optimization time. The optimization time in the k-means algorithm is 0.01 s, and the number of iterations is five. These results indicate the superiority of the k-means algorithm over other clustering algorithms.

Table 11. Comparisons of experimental results.

Algorithm	Time Optimizations (s)	Number of Alterations	Number of Clusters
Enhancing k-means algorithm	0.01	5	5
Traditional k-means algorithm	0.2	8	5
EM algorithm	0.27	22	5
Hierarchical cluster algorithm	0.70	9	5

In this study, we made significant enhancements to the k-means algorithm. These enhancements specifically improve the methods used to determine the number of groups (k) and the selection of initial seeds. We achieved these improvements by utilizing an enhanced GA and optimizing its parameters. We applied this improved GA to predict optimal tourist paths. This prediction is based on tourism objectives derived from social media platforms, considering factors such as popular destinations and peak travel times. Although our approach was specifically implemented for the Port Sudan region of the Red Sea State in Sudan, it can be applied to other regions as well. The suitability of other regions for this approach depends on factors such as the variety of tourist targets, the population characteristics, and the prevalence of relevant social media platforms. However, for accurate and effective implementation, it is essential to conduct a comprehensive study of the region's specific tourist objectives, the population characteristics, and the relevant social networking sites.

5. Conclusions and Future Work

Tourism objectives derived from social media can offer new opportunities for decision support in recommending tourism paths. In this paper, we propose an innovative approach to optimize and classify tourism objectives for recommending the optimal tourism path, using the k-means algorithm and GA. We also integrate various tools for this purpose,

demonstrating the applicability of an improved k-means algorithm and GA for developing tourism path planning. Additionally, we utilize the GIS to implement and visualize efficient social media tourism objectives and display the optimal routes. Our approach is organized as follows: First, the tourism objectives were collected from surveys and social media platforms. Second, the GA was used to enhance the k-means algorithm with a new parameter for clustering tourism objectives. Finally, a comparison and combination were performed with the algorithms currently used in the GIS environment. The following points are recommended:

- ❖ Optimize and classify 12 tourism objectives based on social media platforms to determine the path of tourists.
- ❖ Apply the GA to determine the number of clusters, initial seeds, and the optimal path planning.
- ❖ Optimize and visualize the tourism path planning approach based on the social media tourism objectives.

There is still room for improvement in using ML algorithms to improve social media data. Future work can focus on increasing the objectives and fusing both internal and external objectives in the evaluations of web users.

Author Contributions: Conceptualization, Mohamed A. Damos, Rashad Elhabob, and Jun Zhu; methodology, Weilian Li; software, Elhadi Khalifa; validation, Abubakr Hassan, Mohamed A. Damos, Weilian Li, and Jun Zhu; writing–original draft preparation, Abubakr Hassan; writing–review and editing, Elhadi Khalifa; visualization, Jun Zhu, Abubakr Hassan, and Alaa Hm; funding acquisition, Mohamed A. Damos. Resources, Weilian Li and Esra Ei. All authors have read and agreed to the published version of the manuscript.

Funding: This article was supported by the National Natural Science Foundation of China [grant number 42171397].

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, Mohamed A. Damos, upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Minazzi, R. *Social Media Marketing in Tourism and Hospitality*; Springer: Cham, Switzerland, 2015.
2. Tenemaza, M.; Luján-Mora, S.; De Antonio, A.; Ramirez, J. Improving itinerary recommendations for tourists through metaheuristic algorithms: An optimization proposal. *IEEE Access* **2020**, *8*, 79003–79023. [CrossRef]
3. Lee, J.Y.; Tsou, M.-H. Mapping spatiotemporal tourist behaviors and hotspots through location-based photo-sharing service (Flickr) data. In *Progress in Location Based Services 2018*; Springer: Cham, Switzerland, 2018; pp. 315–334.
4. Wan, L.; Hong, Y.; Huang, Z.; Peng, X.; Li, R. A hybrid ensemble learning method for tourist route recommendations based on geo-tagged social networks. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 2225–2246. [CrossRef]
5. Zhu, J.; Zhang, J.; Zhu, Q.; Li, W.; Wu, J.; Guo, Y. A knowledge-guided visualization framework of disaster scenes for helping the public cognize risk information. *Int. J. Geogr. Inf. Sci.* **2024**, *38*, 1–28. [CrossRef]
6. Aftab, S.; Khan, M.M. Role of social media in promoting tourism in Pakistan. *J. Soc. Sci. Humanit.* **2019**, *58*, 101–113. [CrossRef]
7. Jimenez-Barreto, J.; Sthapit, E.; Rubio, N.; Campo, S. Exploring the dimensions of online destination brand experience: Spanish and North American tourists' perspectives. *Tour. Manag. Perspect.* **2019**, *31*, 348–360. [CrossRef]
8. Ahsini, Y.; Díaz-Masa, P.; Inglés, B.; Rubio, A.; Martínez, A.; Magraner, A.; Conejero, J.A. The Electric Vehicle Traveling Salesman Problem on Digital Elevation Models for Traffic-Aware Urban Logistics. *Algorithms* **2023**, *16*, 402. [CrossRef]
9. Silva, C.E.; César, T.S.; Gomes, I.P.; Silva, J.A.; Wolf, D.F.; Alves, R.; Souza, J.R. Scheduling System for Multiple Self-driving Cars Using K-Means and Bio-inspired Optimization Algorithms. *SN Comput. Sci.* **2023**, *4*, 647. [CrossRef]
10. Gaur, L.; Afaq, A.; Solanki, A.; Singh, G.; Sharma, S.; Jhanjhi, N.; My, H.T.; Le, D.-N. Capitalizing on big data and revolutionary 5G technology: Extracting and visualizing ratings and reviews of global chain hotels. *Comput. Electr. Eng.* **2021**, *95*, 107374. [CrossRef]
11. Hamid, R.A.; Albahri, A.S.; Alwan, J.K.; Al-Qaysi, Z.; Albahri, O.S.; Zaidan, A.; Alnoor, A.; Alamoodi, A.H.; Zaidan, B. How smart is e-tourism? A systematic review of smart tourism recommendation system applying data management. *Comput. Sci. Rev.* **2021**, *39*, 100337. [CrossRef]
12. Li, W.; Zhu, J.; Zhu, Q.; Zhang, J.; Han, X.; Dehbi, Y. Visual attention-guided augmented representation of geographic scenes: A case of bridge stress visualization. *Int. J. Geogr. Inf. Sci.* **2024**, *38*. [CrossRef]

13. Ahmed, M.; Seraj, R.; Islam, S.M.S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics* **2020**, *9*, 1295. [CrossRef]
14. Jahwar, A.F.; Abdulazeez, A.M. Meta-heuristic algorithms for K-means clustering: A review. *PalArch's J. Archaeol. Egypt/Egyptol.* **2020**, *17*, 12002–12020.
15. Huang, J. Design of Tourism Data Clustering Analysis Model Based on K-Means Clustering Algorithm. In *International Conference on Multi-Modal Information Analytics*; Springer: Cham, Switzerland, 2022; pp. 373–380.
16. Yuan, C.; Yang, H. Research on K-value selection method of K-means clustering algorithm. *J* **2019**, *2*, 226–235. [CrossRef]
17. Ikotun, A.M.; Ezugwu, A.E.; Abualigah, L.; Abuhaija, B.; Heming, J. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Inf. Sci.* **2023**, *622*, 178–210. [CrossRef]
18. Yang, Z.; Jiang, F.; Yu, X.; Du, J. Initial Seeds Selection for K-means Clustering Based on Outlier Detection. In Proceedings of the 2022 5th International Conference on Software Engineering and Information Management (ICSIM), Yokohama, Japan, 21–23 January 2022; pp. 138–143.
19. Li, W.; Zhu, J.; Fu, L.; Zhu, Q.; Xie, Y.; Hu, Y. An augmented representation method of debris flow scenes to improve public perception. *Int. J. Geogr. Inf. Sci.* **2021**, *35*, 1521–1544. [CrossRef]
20. Han, M. Research on optimization of K-means Algorithm Based on Spark. In Proceedings of the 2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 24–26 February 2023; pp. 1829–1836.
21. Bahmani, B.; Moseley, B.; Vattani, A.; Kumar, R.; Vassilvitskii, S. Scalable k-means++. *arXiv* **2012**, arXiv:1203.6402. [CrossRef]
22. Chaudhary, M.; Pruthi, J.; Jain, V.K.; Suryakant. A novel squirrel search clustering algorithm for text document clustering. *Int. J. Inf. Technol.* **2022**, *14*, 3277–3286. [CrossRef]
23. Al Shaqsi, J.; Wang, W. Robust Clustering Ensemble Algorithm. *SSRN Electron. J.* **2022**. [CrossRef]
24. Alzyadat, T.; Yamin, M.; Chetty, G. Genetic algorithms for the travelling salesman problem: A crossover comparison. *Int. J. Inf. Technol.* **2020**, *12*, 209–213. [CrossRef]
25. Al-Kaseem, B.R.; Taha, Z.K.; Abdulmajeed, S.W.; Al-Raweshidy, H.S. Optimized energy–efficient path planning strategy in WSN with multiple Mobile sinks. *IEEE Access* **2021**, *9*, 82833–82847. [CrossRef]
26. Chen, J.; Zhang, Y.; Wu, L.; You, T.; Ning, X. An adaptive clustering-based algorithm for automatic path planning of heterogeneous UAVs. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 16842–16853. [CrossRef]
27. Ahmed, A.; Ju, H.; Yang, Y.; Xu, H. An Improved Unit Quaternion for Attitude Alignment and Inverse Kinematic Solution of the Robot Arm Wrist. *Machines* **2023**, *11*, 669. [CrossRef]
28. Hu, F.; Li, Z.; Yang, C.; Jiang, Y. A graph-based approach to detecting tourist movement patterns using social media data. *Cartogr. Geogr. Inf. Sci.* **2019**, *46*, 368–382. [CrossRef]
29. Riaz, M.; Sherani. Investigation of information sharing via multiple social media platforms: A comparison of Facebook and WeChat adoption. *Qual. Quant.* **2021**, *55*, 1751–1773. [CrossRef]
30. Hashimy, S.Q.; Halim, T.S. The Impact of Social Media on Afghanistan's Tourism Industry: A Roadmap for the Future in the Internet Highway. *Law Soc. Policy Rev.* **2023**, *1*, 17–50.
31. Sakas, D.P.; Reklitis, D.P.; Terzi, M.C.; Vassilakis, C. Multichannel digital marketing optimizations through Big Data Analytics in the tourism and Hospitality Industry. *J. Theor. Appl. Electron. Commer. Res.* **2022**, *17*, 1383–1408. [CrossRef]
32. Kim, J.; Kang, Y. Automatic classification of photos by tourist attractions using deep learning model and image feature vector clustering. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 245. [CrossRef]
33. Chen, Y.; Zheng, X.; Fang, Z.; Yu, Y.; Kuang, Z.; Huang, Y. Research on optimization of tourism route based on genetic algorithm. *J. Phys. Conf. Ser.* **2020**, *1575*, 012027. [CrossRef]
34. Kamsing, P.; Torteeka, P.; Yooyen, S.; Yenpiem, S.; Delahaye, D.; Notry, P.; Phisannupawong, T.; Channumsin, S. Aircraft trajectory recognition via statistical analysis clustering for Suvarnabhumi International Airport. In Proceedings of the 2020 22nd International Conference on Advanced Communication Technology (ICACT), Pyeongchang, Republic of Korea, 16–19 February 2020; pp. 290–297.
35. Dadashpour Moghaddam, M.; Ahmadzadeh, H.; Valizadeh, R. A GIS-based assessment of urban tourism potential with a branding approach utilizing hybrid modeling. *Spat. Inf. Res.* **2022**, *30*, 399–416. [CrossRef]
36. Zhou, X.; Xu, C.; Kimmons, B. Detecting tourism destinations using scalable geospatial analysis based on cloud computing platform. *Comput. Environ. Urban Syst.* **2015**, *54*, 144–153. [CrossRef]
37. Wang, H.; Yan, J. Effects of social media tourism information quality on destination travel intention: Mediation effect of self-congruity and trust. *Front. Psychol.* **2022**, *13*, 1049149. [CrossRef]
38. Sarkar, S.K.; George, B. Social media technologies in the tourism industry: An analysis with special reference to their role in sustainable tourism development. *Int. J. Tour. Sci.* **2018**, *18*, 269–278. [CrossRef]
39. Tahir, M.; Tubaishat, A.; Al-Obeidat, F.; Shah, B.; Halim, Z.; Waqas, M. A novel binary chaotic genetic algorithm for feature selection and its utility in affective computing and healthcare. *Neural Comput. Appl.* **2020**, *34*, 1–22. [CrossRef]
40. Damos, M.A.; Zhu, J.; Li, W.; Hassan, A.; Khalifa, E. A novel urban tourism path planning approach based on a multiobjective genetic algorithm. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 530. [CrossRef]
41. Pizzuti, C.; Procopio, N. A k-means based genetic algorithm for data clustering. In Proceedings of the International Joint Conference SOCO'16-CISIS'16-ICEUTE'16, San Sebastián, Spain, 19–21 October 2016; Proceedings 11, pp. 211–222.

42. Tabianan, K.; Velu, S.; Ravi, V. K-means clustering approach for intelligent customer segmentation using customer purchase behavior data. *Sustainability* **2022**, *14*, 7243. [CrossRef]
43. Ghezelbash, R.; Maghsoudi, A.; Shamekhi, M.; Pradhan, B.; Daviran, M. Genetic algorithm to optimize the SVM and K-means algorithms for mapping of mineral prospectivity. *Neural Comput. Appl.* **2023**, *35*, 719–733. [CrossRef]
44. Zubair, M.; Iqbal, M.A.; Shil, A.; Chowdhury, M.; Moni, M.A.; Sarker, I.H. An improved K-means clustering algorithm towards an efficient data-driven modeling. *Ann. Data Sci.* **2022**, *9*, 1–20. [CrossRef]
45. Daviran, M.; Ghezelbash, R.; Niknezhad, M.; Maghsoudi, A.; Ghaeminejad, H. Hybridizing K-means clustering algorithm with harmony search and artificial bee colony optimizers for intelligence mineral prospectivity mapping. *Earth Sci. Inform.* **2023**, *16*, 2143–2165. [CrossRef]
46. Sajidha, S.; Desikan, K.; Chodnekar, S.P. Initial seed selection for mixed data using modified k-means clustering algorithm. *Arab. J. Sci. Eng.* **2020**, *45*, 2685–2703. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

Smart Urban Cadastral Map Enrichment—A Machine Learning Method

Alireza Hajiheidari [1], Mahmoud Reza Delavar [2,*] and Abbas Rajabifard [3]

[1] GIS Department, School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran 1417614411, Iran; aliz.hajiheidari@ut.ac.ir
[2] Center of Excellence in Geomatic Engineering in Disaster Management, Land Administration in Smart City Laboratory, School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran 1417614411, Iran
[3] The Centre for Spatial Data Infrastructures and Land Administration, Department of Infrastructure Engineering, University of Melbourne, Melbourne, VIC 3010, Australia; abbas.r@unimelb.edu.au
* Correspondence: mdelavar@ut.ac.ir

Abstract: Enriching and updating maps are among the most important tasks of any urban management organization for informed decision making. Urban cadastral map enrichment is a time-consuming and costly process, which needs an expert's opinion for quality control. This research proposes a smart framework to enrich a cadastral base map using a more up-to-date map automatically by machine learning algorithms. The proposed framework has three main steps, including parcel matching, parcel change detection and base map enrichment. The matching step is performed by checking the center point of each parcel in the other map parcels. Support vector machine and random forest classification algorithms are used to detect the changed parcels in the base map. The proposed models employ the genetic algorithm for feature selection and grey wolf optimization and Harris hawks optimization for hyperparameter optimization to improve accuracy and performance. By assessing the accuracies of the models, the random forest model with feature selection and grey wolf optimization, with an F1-score of 0.9018, was selected for the parcel change detection method. Finally, the detected changed parcels in the base map are deleted and relocated automatically with corresponding parcels in the more up-to-date map by the affine transformation.

Keywords: urban cadastral map enrichment; spatial data integration; spatial data matching; change detection in urban cadastral maps; feature selection; grey wolf optimization algorithm; Harris hawks optimization algorithm

1. Introduction

Cities are becoming more complex day to day due to increments in population, urbanization and the pace of construction and change [1]. The vast extents, high densities and fast changes in cities and urban areas make measuring, managing, monitoring, mapping and modeling urban changes more challenging and time-consuming for organizations and researchers [1]. The growth of urbanization causes intensive use of land. The faster growth of the physical extent of urban areas compared to their population causes more land to be used for urban development [2]. Hence, efficient land administration and parcel-based land development modeling play an important role in sustainable urban growth [3,4]. Land Administration Systems (LASs) in cities are for intelligent land and urban space management [5]. Land information is required for different LAS components, which are land development, land use, land value and land tenure [5]. Urban cadastral maps as a part of land information are large-scale maps that contain dimensions, shapes, spatial relationships and different attributes of information about land parcels [6].

Enrichment of urban cadastral maps helps organizations to make accurate and timely decisions for urban management [7]. Urban cadastral map enrichment and checking for the

accuracy and reliability of data are important duties of urban management organizations [8]. Urban cadastral maps are enriched with the help of legal documents for geometric changes, incorporating new information and data, or the removal of inaccurate data [8]. Enrichment and updating of urban cadastral maps are as important as their production and pose different challenges [9]. Map enrichment is a parcel-based time-consuming process due to the large number of parcels and the large amount of information in urban areas and needs visual parcel-to-parcel checking and a high level of expertise in the different steps of enrichment, including matching, change detection and base map enrichment. Hence, detecting changes in a map with more up-to-date maps from other sources automatically by machine learning algorithms can reduce the time and cost of map enrichment. Machine learning methods such as random forest are good methods with computational advantages for large-scale urban cadastral maps and spatial data analyses [4]. Supervised machine learning methods are some of the change detection techniques that require training data for classification [10].

Map enrichment has been performed using different methods and data and for different usages. In some research, historical and analogue cadastral maps have been used to determine ownership boundaries and in the renewal of cadastral maps [6,9,11]. Data and map integration are other methods for map enrichment. Integration of different web maps [12] and enriching a base map with a more up-to-date map, aerial images, orthophotos and remote sensing (RS) imagery [7,13–17] are some of the spatial data fusion and integration methods which have been previously investigated.

Artificial intelligence methods, including machine learning and deep learning, have recently been used very much for map enrichment; updating; object, boundary and change detection; and spatial data fusion [14,15,18–20]. In [18], a combination of data-level fusion and feature-level fusion was employed for natural object detection by multi-source geospatial data and deep learning. Deep learning methods have been implemented to align and update cadaster maps with satellite images [19], detect visible land boundaries automatically with aerial images and revise existing cadastral maps [20]. In [21], a comparison between manual approaches and machine learning algorithms for extracting visible cadaster boundaries from satellite images in rural and urban areas was made, and the authors showed that machine learning algorithms have lower costs, require less time and achieve a higher accuracy than manual approaches.

Most previous research has been focused on aerial or satellite images to enrich base maps. Most of them have been employed in small areas, or map features have been enriched case by case, and less attention has been given to enriching base maps with more recent maps in a vector space in vast, dense and complex urban areas. In addition, automatic change detection and map modification with an acceptable accuracy have been less investigated. Hence, the contribution of this research is to propose an intelligent framework to enrich cadastral parcels in a base map of an urban area with a more up-to-date map automatically.

The remaining parts of this paper are as follows: Section 2 discusses the methodology of the research, the proposed methods and their evaluation. Section 3 presents the study area, the employed data and the implementation of the proposed method. Section 4 constitutes the discussion, and Section 5 concludes the paper and recommends some directions for future research.

2. Proposed Methodology

This research is aimed at reaching an intelligent framework for enriching land parcels in urban base maps. Two urban maps were employed in this research: a base map (a Cadaster Department map, a map at 1:2000 scale in the UTM (Zone 39N) projection system, produced in 2002) that was to be enriched and a more up-to-date urban map (a Tehran Municipality map, a map at 1:1000 scale in the UTM (Zone 39N) projection system, produced in 2014) that was used to enrich the base map. Parcels in each map were target features in this research, and the focus was only on enriching the geometry

of the parcels. Due to limitations on data access, other information for each parcel, such as registration, legal and descriptive information, was not investigated. This framework has three main steps: matching, parcel change detection and base map enrichment. In this research, after preprocessing the data, including format and scale homogenization and topology checking, a matching algorithm was employed to find the corresponding parcels for each parcel in each of the maps. Then, 38 different geometric, topologic and statistical parameters were calculated for each parcel in each of the maps to be used in support vector machine (SVM) and random forest (RF) machine learning algorithms for detecting changes in the base map intelligently. There are different methods to increase the accuracy of models, including data addition, dimension reduction, regularization, missing value and outlier management, feature engineering, feature selection, ensemble learning, and hyperparameter optimization. Missing value and outlier management, feature engineering, feature selection, ensemble learning (using random forest algorithms), and hyperparameter optimization were employed in this research in the models' training steps. In training the models, feature selection and hyperparameter optimization were used to increase the accuracy of change detection. Finally, the changed parcels that were detected in the previous step were replaced automatically by new parcels in the more up-to-date map. In this step, each parcel in the more up-to-date map that corresponds to a changed parcel in the base map is located on the base map by the unchanged neighbors of the parcel.

2.1. Matching

In spatial datasets, the matching process is the identification of corresponding features in two or more different datasets in the same spatial coverage, where the corresponding items represent the same feature in the real world [22]. Matching is one of the main steps in enriching and integrating heterogeneous spatial data, which is performed to add new data, maintain similar and unchanged spatial data, remove additional data, and reduce differences and contradictions between data and information in the datasets [23,24].

Since the data used in this research are parcels, if these parcels are used and the intersection or sharing of borders between parcels of the first map and the second map is examined, the neighbors of the parcels are also considered as the corresponding parcels. This causes errors in detecting corresponding parcels. Hence, to solve this problem, according to Figure 1, gravity centers of parcels are calculated and the matching process is performed by these gravity centers.

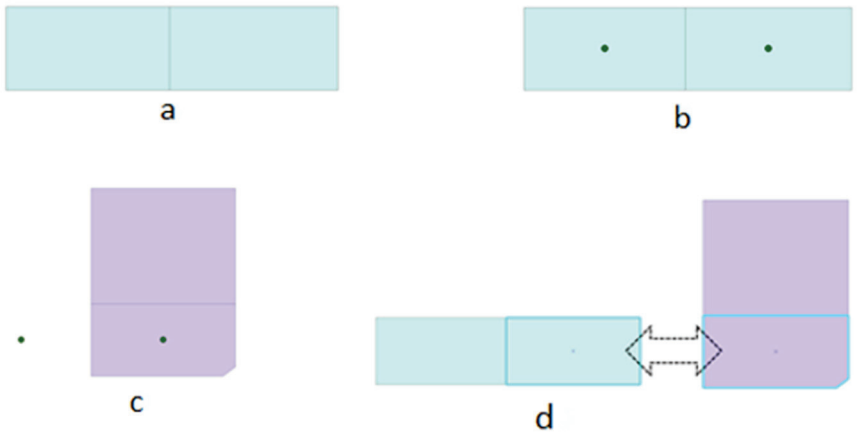


Figure 1. Parcel matching. Green parcels: base map parcels, green points: gravity centers of base map parcels, purple parcels: second map parcels. (**a**) Base map parcels. (**b**) Calculated gravity centers of parcels. (**c**) Transfer of center points to second map. (**d**) Detection of the corresponding parcels.

The gravity center is the center of mass of a parcel that may fall inside or outside the parcel. The coordinate of the gravity center point ($C_x \cdot C_y$) of a non-self-intersecting parcel is calculated by Equations (1) and (2) [25].

$$C_x = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (1)$$

$$C_y = \frac{1}{6A}\sum_{i=0}^{n-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (2)$$

where (x_i, y_i) is the coordinate of the i vertex of the polygon, (x_n, y_n) and (x_0, y_0) are the same, and A is the signed area of the polygon that is calculated by the Shoelace formula (Gauss's area formula) by Equation (3) [25].

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i) \quad (3)$$

In this research, to identify the corresponding parcels in the base map and the second map, the gravity centers of the parcels in the base map were calculated. Then, parcels from the second map where these centers were located were considered as corresponding parcels for each parcel in the base map. In order to consider all the changes for the parcels in the base map compared to the more up-to-date map, this matching must be performed in both maps, which means that this matching should be implemented in parcels in the second maps too. Figure 2 illustrates the matching method in this research.

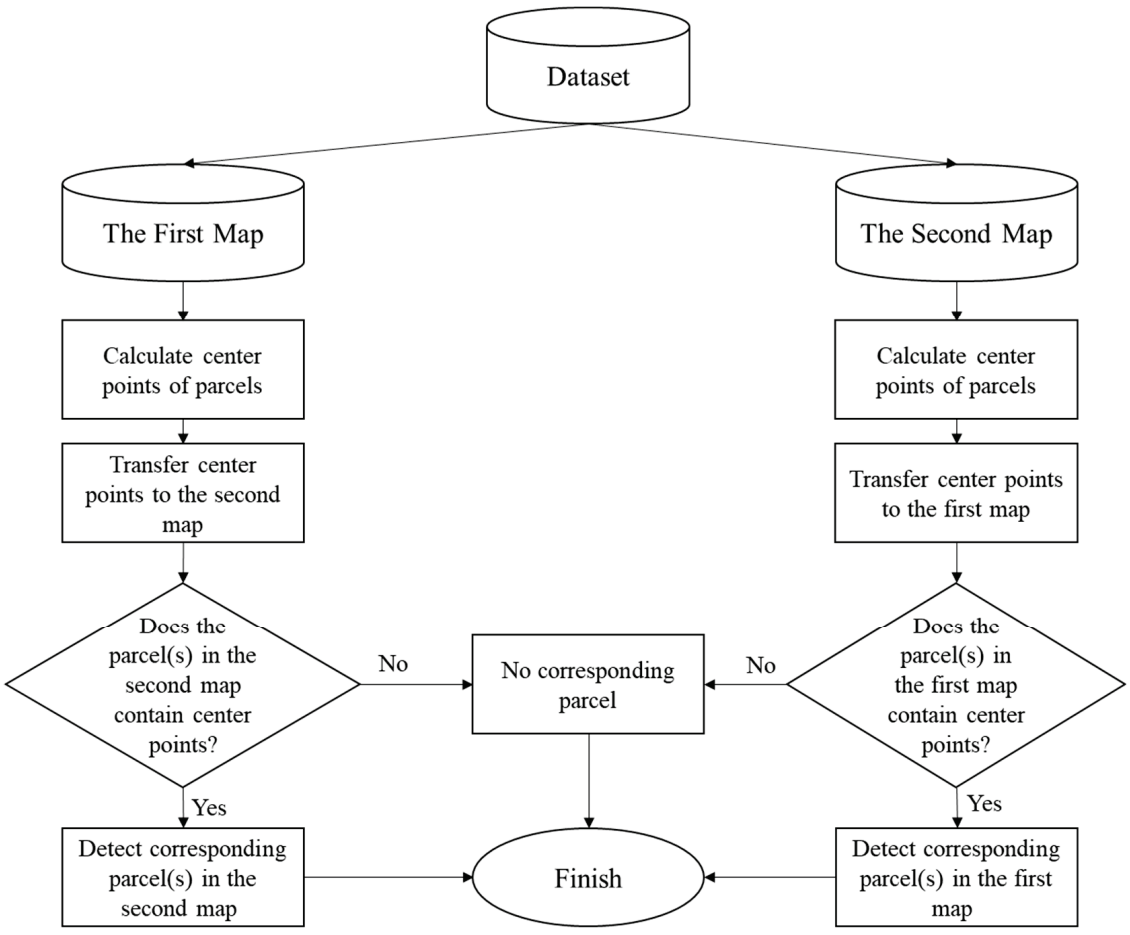


Figure 2. Parcel matching process.

It is necessary to implement the matching algorithm in both maps to reach all these relations. Based on these relations, some changes in the base map can be detected. To better train and evaluate the change detection model, training and test data should be selected from all categories that exist in the matching process. Table 1 and Figure 3 show the different relations in the matching process.

Table 1. Matching relations.

No.	Relation	Number of Relations in Figure 3	Description
1	1:0	2	The parcel is deleted in the second map
2	0:1	4	A parcel is created in the second map
3	1:n	8 → 10,11	The parcel is split in the second map
4	n:1	5,6,7 → 9	The parcel is merged in the second map
5	1:1	1 → 3	There is a corresponding parcel in the second map

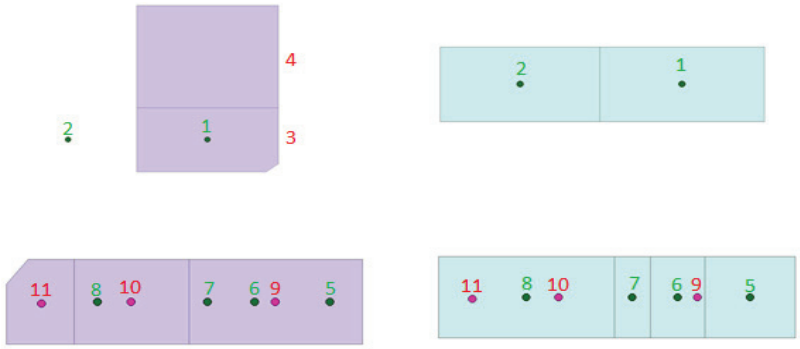


Figure 3. Parcel matching. Green parcels: base map parcels, green points and numbers: gravity centers of base map parcels, purple parcels: second map parcels, red points and numbers: gravity centers of second map parcels.

Because of the effect of the number of vertices of a polygon in determining the center of gravity of that polygon, the validation of matching is performed for parcels with different numbers of vertices. Figure 4 shows the effect of the number of vertices on determining the center of gravity and their effect on incorrect matching.

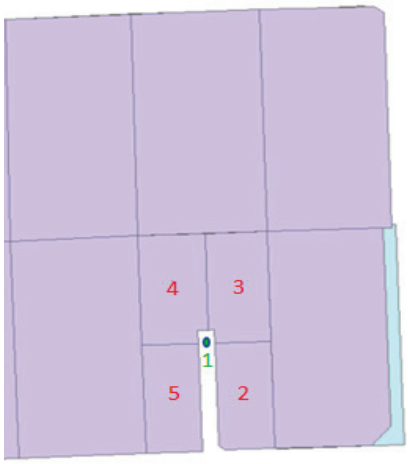


Figure 4. Effect of number of vertices in determining the center of gravity. Green parcels: base map parcels, green points and numbers: gravity centers of base map parcels, purple parcels: second map parcels, red numbers: the second map parcels that corresponds to parcel number one from the base map.

In Figure 4, parcel number one corresponds to parcels number two, three, four and five, and it is the union of parcels in the second map, and its center of gravity is point number one, which is not located inside any parcels of the second map. If the matching is not performed from the second map to the first, this feature will not be correctly matched, or, if the parcel is not separated, the parcel will not be matched correctly.

2.2. Classification and Parcel Change Detection Modeling

The purpose of this step is to find parcels that have changed in the base map. For this purpose, after finding the corresponding parcels and by checking the geometric, topologic and statistical parameters, the changed parcels are identified by classification with support vectors machine (SVM) and random forest (RF) methods. To increase accuracy and reduce computational complexity, feature selection was used and hyperparameters were optimized by metaheuristic algorithms.

The geometric, topologic and statistical parameters are important parameters for features in maps. Hence, 38 different geometric, topologic and statistical parameters were considered for comparing parcels in the two maps [26,27]. To check the correlation between the parameters and features and prevent noise in the models, a correlation matrix was calculated between the calculated parameters (features) and then the parameters were selected for modeling. By checking the correlation of features, 19 dependent features were removed from the list of features of the parcels, and a total of 19 features with low dependencies were considered for each parcel to be used in the modeling. Table 2 shows a list of all the geometric, topologic and statistical parameters considered and the ones that were selected after checking the correlation of the features.

Table 2. Geometric, topologic and statistical parameters and the ones that were selected after the correlation checking of the features.

No.	Parameters	Selected	No.	Parameters	Selected
1	Area	Yes	20	MBB APodY1	Yes
2	Area Geodesic	No	21	MBB APodX2	No
3	Perimeter	No	22	MBB APodY2	No
4	Perimeter Geodesic	No	23	MBB Orientation	Yes
5	Centroid X	Yes	24	MBB Perimeter	No
6	Centroid Y	Yes	25	MBB Area	Yes
7	Start X	No	26	Number of Polygon Neighbors	No
8	Start Y	No	27	Minimum Length	No
9	Mid X	Yes	28	Maximum Length	No
10	Mid Y	Yes	29	Average Length	Yes
11	Node Count	Yes	30	Sum Length	No
12	Minimum Extent X	No	31	Standard Division Length	No
13	Minimum Extent Y	Yes	32	Variance Length	Yes
14	Maximum Extent X	No	33	Minimum Node	No
15	Maximum Extent Y	Yes	34	Maximum Node	Yes
16	Polygon Main Angle	Yes	35	Average Node	Yes
17	Minimum Bounding Box (MBB) Width (Convex)	Yes	36	Sum Node	Yes
18	MBB Length	No	37	Standard Division Node	No
19	MBB Antipodal Pairs (APod) X1	Yes	38	Variance Node	No

2.2.1. Support Vector Machine

The support vector machine (SVM) is an algorithm that searches for an optimum hyperplane for linear separation of categories by mapping the training data in a non-linear way to a higher dimension [28]. This algorithm uses support vectors and edges to find this

hyperplane. The advantages of this algorithm are its ability to model linear, non-linear and complex problems, its high accuracy, and its low probability of data overfitting. This algorithm is suitable for classification and numerical predictions [28–30]. If the data are located above the separating hyperplane, they are labeled 1; otherwise, they are labeled -1. The hyperplane equation corresponds to Equation (4) [31].

$$y_i(W \cdot X_i + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0, \forall i \tag{4}$$

where W is the normal vector that is perpendicular to the hyperplane, b is a deviation number and ε (an error margin parameter) specifies the extent of violation of the determined boundary for each data. ε is used to prevent the adverse effect of noise data on decision-making boundaries and overfitting. Equation (5) is for optimization, and the new data are classified by a trained SVM and the Lagrange formulation according to Equation (6) [31].

$$\min_{w,b,\varepsilon} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \varepsilon_i \tag{5}$$

$$d\left(X^T\right) = \sum_{i=1}^{l} y_i \alpha_i X_i X^T + b_0 \tag{6}$$

In Equation (5), C is the variable that regularizes the allowable value of the model for misclassification [31]. In Equation (6), y_i is the label of the class for X_i and X^T is a new data. The Lagrangian coefficients α_i and b_0 are numerical parameters from the optimization in Equation (5) [31].

2.2.2. Random Forest

There are various methods to improve performance and classification accuracy. The ensemble learning method is one of these methods, and the desired model consists of several classifiers. There is a high possibility of error if basic classifiers are used, but in ensemble methods, if more than half of the categories are wrong, the final output will contain errors [31]. Boosting and bagging are some methods of ensemble learning methods [32]. In bagging algorithms, models are trained simultaneously, in parallel and independently. To create each model, data are randomly selected from the main dataset, while there will be a possibility of having some data in all or some models. Finally, the result will be equal to the vote of the majority of the models. One of the bagging algorithms is the random forest (RF) algorithm [31–33]. In the RF algorithm. the classifiers used in the ensemble method are of the decision tree algorithms, and, finally, the best class is predicted based on voting. This algorithm performs better in comparison with boosting algorithms and has better performance in datasets with errors or outliers [31,34].

2.2.3. Genetic Algorithm

The genetic algorithm (GA) is one of the evolutionary and meta-heuristic computing methods, which is based on Darwin's theory of biological evolution [35]. This algorithm has been used in feature selection for parcel change detection. At first, an initial population that includes a set of random rules is created. Each rule in this algorithm is coded with the help of a K-bit chromosome, where K is the number of values in that rule. Then, a new population of meritorious rules is generated based on the theory of survival of the fittest. New generations and children are created by genetic operators, such as selection, intersection and mutation, from the appropriate rules. The process of producing new populations by the populations of the previous stage continues until the termination condition is fulfilled. This termination condition can be the fulfillment of the fitness threshold of each of the rules in the final population, the production of a certain number of new populations, or the creation of several populations without improving the rejection of the rules fitness [31,36,37].

2.2.4. Grey Wolf Optimization

Grey wolf optimization (GWO) is one of the meta-heuristic algorithms based on the collective intelligence and social behavior of grey wolves during hunting [38]. Due to the examination of the grey wolf algorithm with some other meta-heuristic algorithms and the presentation of competitive results, as well as proof of the application of the grey wolf algorithm for engineering problems with an unknown search space [38], this method was selected as one of the optimization algorithms in this research. The hunting process in grey wolves includes searching and identifying prey, surrounding the prey, and attacking the prey [38]. The wolves in each pack are divided into one of four kinds: Alpha, Beta, Delta and Omega. Optimization is performed by three wolves: Alpha, Beta and Delta. In the grey wolf optimization algorithm, Alpha is the best solution, and Beta and Delta are the second and third best solutions, respectively. Equations (7)–(10) present the grey wolf optimization algorithm [38].

$$\vec{D} = \left| \vec{C} \cdot \vec{X_p}(t) - \vec{X}(t) \right| \tag{7}$$

$$\vec{X}(t+1) = \vec{X_p}(t) - \vec{A} \cdot \vec{D} \tag{8}$$

$$\vec{A} = 2\vec{a} \cdot \vec{r_1} - \vec{a} \tag{9}$$

$$\vec{C} = 2\vec{r_2} \tag{10}$$

where t is the iteration number, A and C are the coefficient vectors, X_P is the prey location vector, X is the grey wolf location vector, and r_1 and r_2 are random vectors in the range from zero to one and decrease linearly from two to zero during iterations. The new positions of each wolf in the grey wolf algorithm are calculated based on the best positions, which are the positions of the Alpha, Beta and Delta wolves. Equations (11)–(13) present the update of these locations [38].

$$\vec{D_\alpha} = \left| \vec{C_1} \cdot \vec{X_\alpha} - \vec{X} \right|, \vec{D_\beta} = \left| \vec{C_2} \cdot \vec{X_\beta} - \vec{X} \right|, \vec{D_\delta} = \left| \vec{C_3} \cdot \vec{X_\delta} - \vec{X} \right| \tag{11}$$

$$\vec{X_1} = \vec{X_\alpha} - \vec{A_1} \cdot (\vec{D_\alpha}), \vec{X_2} = \vec{X_\beta} - \vec{A_2} \cdot (\vec{D_\beta}), \vec{X_3} = \vec{X_\delta} - \vec{A_3} \cdot (\vec{D_\delta}) \tag{12}$$

$$\vec{X}(t+1) = \frac{\vec{X_1} + \vec{X_2} + \vec{X_3}}{3} \tag{13}$$

2.2.5. Harris Hawks Optimization

The Harris hawks optimization (HHO) algorithm is a population-based algorithm that is inspired by the chase behavior of Harris hawks in surprising their prey [39]. This algorithm has been used in this research due to the comparison of the Harris hawks optimization algorithm with other nature-inspired optimization algorithms in several engineering problems and the promising and sometimes competitive results of the Harris hawks optimization algorithm with respect to other meta-heuristic algorithms [39]. The Harris hawks algorithm has three main stages of exploration: transition from exploration to exploitation and exploitation (extraction). In the exploration phase, the location vectors of the hawks are calculated by Equations (14) and (15) [39].

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \tag{14}$$

$$X_m(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t) \tag{15}$$

where $q \geq 0.5$ is for situations when the hawks are randomly located in trees and waiting for the prey and $q < 0.5$ is for cases when the hawks' locations are based on other hawk's

locations. In addition, $X(t+1)$ is the location vector in the i_{th} iteration, $X_{rabbit}(t)$ is the position of the prey, $X(t)$ is the current location vector of the hawks, X_{rand} is the location of a random hawk in the current population, X_m is the average location of the current population of hawks, and $r_1.r_2.r_3.r_4$ as well as q are random numbers between zero and one, which are changed and updated in each iteration. LB and UB represent the upper and lower bounds for the variables. N is the total number of hawks, and $X_i(t)$ is the location of each hawk in the i iteration.

To transfer from the exploration stage to the exploitation (extraction) stage, it is necessary to calculate the energy of the prey. This energy is calculated by Equation (16) [39].

$$E = 2E_0(1 - \frac{t}{T}) \tag{16}$$

where E is the energy of the prey to escape, T is the maximum number of repetitions and E_0 is the initial energy of the prey. E_0 has a value between negative one and positive one which changes in each iteration. This energy decreases from one to negative one during the changes.

At the exploitation stage, Harris hawks have four methods of attacking prey based on their prey escape behavior and their pursuit method. These methods are soft besiege, hard besiege, soft besiege with progressive rapid dives, and hard besiege with progressive rapid dives [39]. The Equations (17) and (18) are related to soft besiege [39].

$$X(t+1) = \Delta X(t) - E|JX_{rabbit}(t) - X(t)| \tag{17}$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \tag{18}$$

The values of $\Delta X(t)$ are the difference between the prey location and the current location in repetition t, $J = 2(1 - r_5)$, and r_5 is a random number between zero and one. The value of J represents the power of the prey's random jump in all jumps. which changes randomly in each iteration.

The location of each hawk is updated by Equation (19) in hard besiege [39].

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)| \tag{19}$$

In the soft besiege with progressive rapid dives method, first, the hawks' location and next movements are determined by Equation (20) and a soft besiege. Then, ac-cording to Equations (21) and (22), the hawks will dive towards the prey, and, finally, their locations in soft besiege are updated by Equation (23) [39].

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)| \tag{20}$$

$$Z = Y + S \times LF(D) \tag{21}$$

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}} \tag{22}$$

$$X(t+1) = \begin{cases} Y & if \quad F(Y) < F(X(t)) \\ Z & if \quad F(Z) < F(X(t)) \end{cases} \tag{23}$$

where D is the dimension of the problem, S is a random vector with size 1*D and LF is the flight function calculated by Equation (22). U and v are random values between zero and one, and beta has a constant value of 1.5.

In hard besiege with progressive rapid dives, the locations of hawks in hard besiege are updated by Equation (23), with the difference that the Y and Z values are calculated by Equations (24) and (25) [39].

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X_m(t)| \tag{24}$$

$$Z = Y + S \times LF(D) \tag{25}$$

2.2.6. Validation

To validate the modeling and to calculate the accuracy of parcel change detection, the root mean square error (*RMSE*), overall accuracy (accuracy), *precision*, *recall* and *F1-score* were used in this research. According to Equation (26), the RMSE was used to calculate loss functions [40]. The confusion matrix is shown in Table 3 [41].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y)^2} \tag{26}$$

Table 3. Confusion matrix.

	Detected Positive	**Detected Negative**
Actual Positive	True positive (TP)	False negative (FN)
Actual Negative	False positive (FP)	True negative (TN)

According to the confusion matrix, *accuracy*, *precision*, *recall* and *F1-score* are calculated by Equations (27)–(30) [42].

$$Precision = \frac{TP}{TP + FP} \tag{27}$$

$$Recall = \frac{TP}{TP + FN} \tag{28}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{29}$$

$$F1\text{-}score = 2\frac{Precision \times Recall}{Precision + Recall} \tag{30}$$

Table 4 shows the confusion matrix for the parcel change detection modeling. This matrix indicates the identification of changes in parcels correctly or incorrectly [42].

Table 4. Confusion matrix for parcel change detection modeling.

Parcels	**Actual Change**	**Actual Unchanged**
Change Detection Positive	True positive (TP)	False positive (FP)
Change Detection Negative	False negative (FN)	True negative (TN)

2.2.7. The Proposed Methodology

The proposed model contains SVM and RF classification integrated with the GA for feature selection and the GWO or HHO algorithms for hyperparameter optimization in each iteration of training the models to improve their accuracy.

2.3. Base Map Enrichment

After detecting the changed parcels in the base map, the process of automatic enrichment of the base map (the Cadaster Department map) was performed with the more up-to-date map (the Tehran Municipality map), as shown in Figure 5. First, the detected changed parcels were removed from the base map. In the second step, for each removed parcel in the base map, the nearest neighboring parcels with the corresponding parcel in the more up-to-date map were identified, and then, in the third step, the transformation parameters between the neighbors of that parcel were calculated from the most up-to-date map to the base map. Then, in the fourth step, the vertices of the desired parcel in the updated map were transferred to the base map by the calculated transformation parameters. Finally, in the fifth step, the base map was checked against topologic errors to evaluate the enrichment.

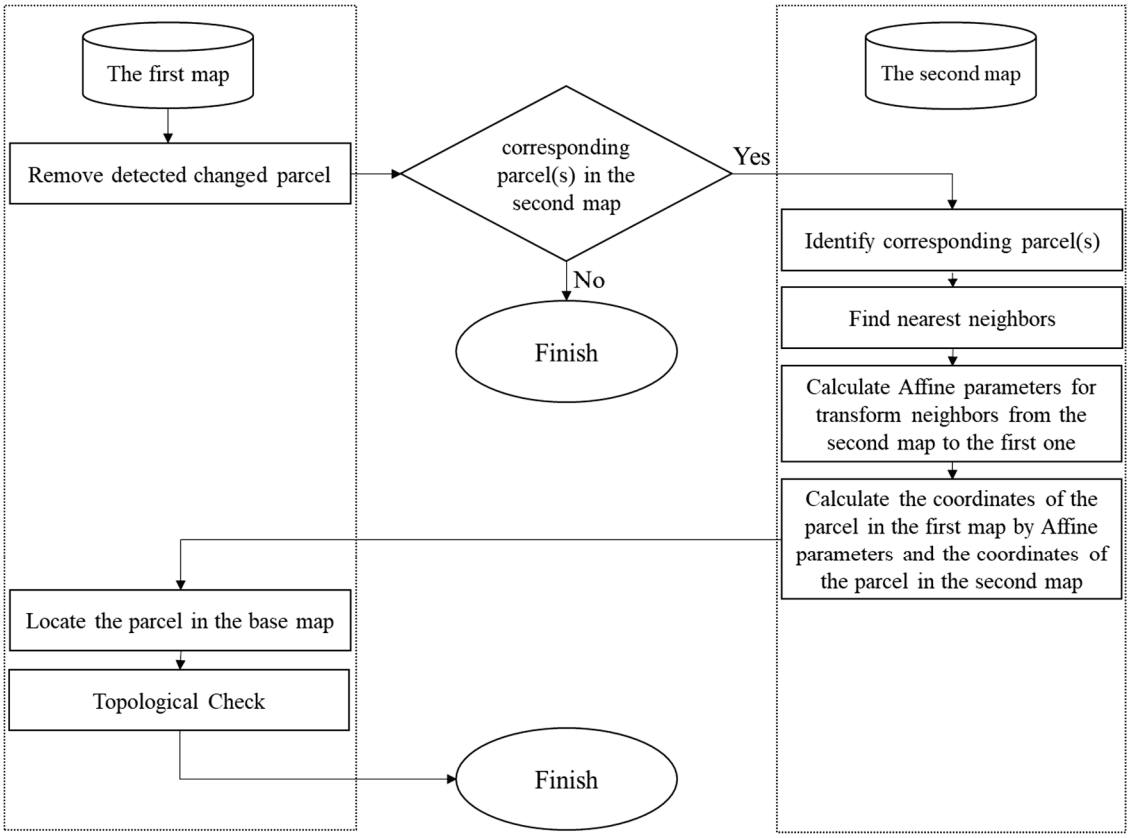


Figure 5. Base map enrichment with the more up-to-date map.

The affine transformation was used in this research to transform between the maps and enrich the base map due to the two-dimensional maps and features [43]. Equations (31) and (32) show the affine transformation. The coordinates of the vertices of the parcels were used in the equations to find the parameters or the transferred coordinates.

$$x' = ax - by + c \tag{31}$$

$$y' = bx + ay + d \tag{32}$$

where x and y are the initial coordinates, y' and x' are the transferred coordinates. and c and d are the transfer factors. a and b are calculated by Equations (33) and (34) [43].

$$a = s\cos\alpha \tag{33}$$

$$b = -s\sin\alpha \tag{34}$$

where s is the scale factor and alpha is the rotation angle, which are calculated from Equations (35) and (36) [43].

$$s = \sqrt{a^2 + b^2} \tag{35}$$

$$\alpha = \tan^{-1}(-\frac{b}{a}) \tag{36}$$

3. Implementation

3.1. Study Area

The study area is the third zone of Tehran Municipality District Six. The capital of Iran, Tehran, consists of 22 districts, 125 zones and 355 neighborhoods. District Six is one of the oldest districts of Tehran Municipality, which is geographically located in the center of Tehran. This district has an area of 2138 square kilometers (3.3% of the total area of Tehran),

a population of 250,000, six zones and 14 neighborhoods [44]. The third zone of Tehran Municipality District Six is located in the middle east of this area. This zone is one of the oldest, most highly populated zones in Tehran. Hence, this zone was selected as the study area, which is illustrated in Figure 6.

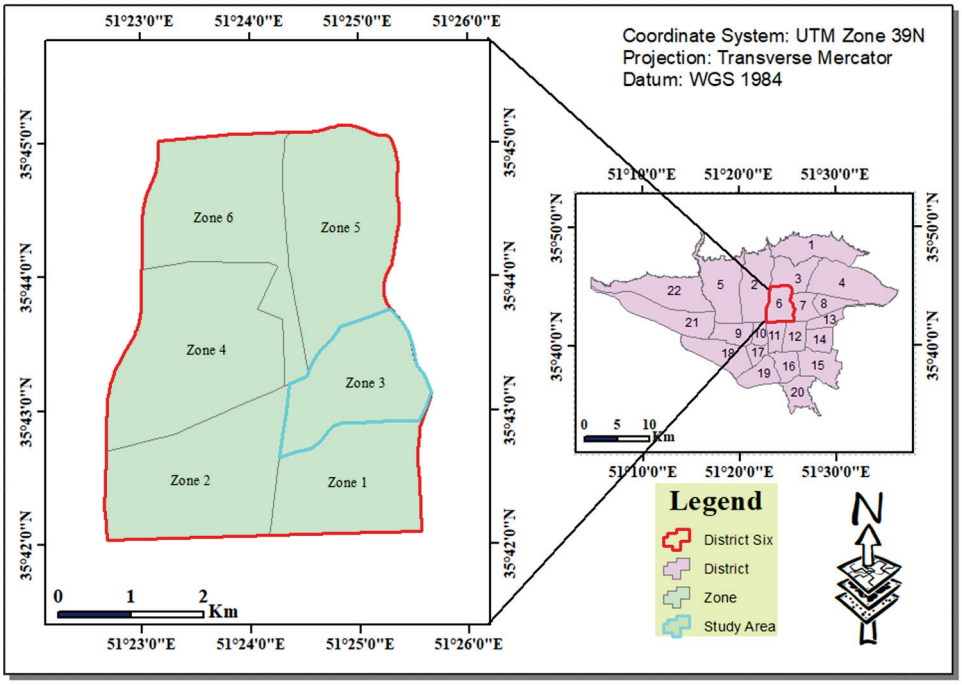


Figure 6. The study area.

3.2. Data

The data employed include a parcel map of the Cadastral Department and a parcel map of Tehran Municipality.

3.2.1. Iranian Cadaster Deaprtment Dataset

The Iranian Cadaster Department is under the auspices of the Iran Deeds and Property Registration Organization, which prepares documents for property registration records. One of the maps in the Cadaster Department is the base map produced in 2002 by the National Cartography Centre of Iran (NCC) through photogrammetry, which has been enriched and updated with new maps and other organizations' maps. The map is at 1:2000 scale and is in accordance with the UTM (Zone 39N) projection system. The map does not include any registered parcels, and there is another database for the registered parcels in the Cadaster Department. Hence, enriching the base map and relocating parcels from the other map did not change any registered parcel.

3.2.2. Tehran Municipality Dataset

Tehran Municipality is the main organization that manages the city of Tehran and designs new plans for urban development. Hence, it needs to have the latest updated urban maps for effective urban management. In addition, any construction and geometric or land use changes in buildings in Tehran need the municipality's approval. The Tehran Municipality map was produced in 2014 by the photogrammetric method and has been enriched and updated since then. The map is at 1:1000 scale and is in accordance with the UTM (Zone 39N) projection system.

3.3. Preprocessing of the Employed Data

The mapping data of different organizations can be heterogeneous and incompatible due to differences in methods and standards of map preparation, maintenance, updating,

data preparation time, format, scale and applications [45]. Metadata play a very important role in data management and data quality-control processes [46].

To integrate and compare the datasets, the two maps were converted to the same format (Shapefile (Esri Format)) and the parcel topology was checked. Then, both of the datasets were converted to 1:2000 scale. Hence, the Tehran Municipality map was generalized from 1:1000 to 1:2000 scale by polygon simplification and the Douglas–Peucker algorithm with 0.1 m tolerance. This method preserved the polygons' shapes and removed vertices with more than 0.1 m perpendicular distance to the new line added after the vertices' removal [47–49]. Finally, parcel geometry was checked and modified.

Figure 7 illustrates parcels in the Cadaster Department map and the Tehran Municipality map in the third zone of District Six after the preprocesses. The differences in the parcels and block numbers make it clear that some changes in the Cadaster Department map are essential for its enrichment.

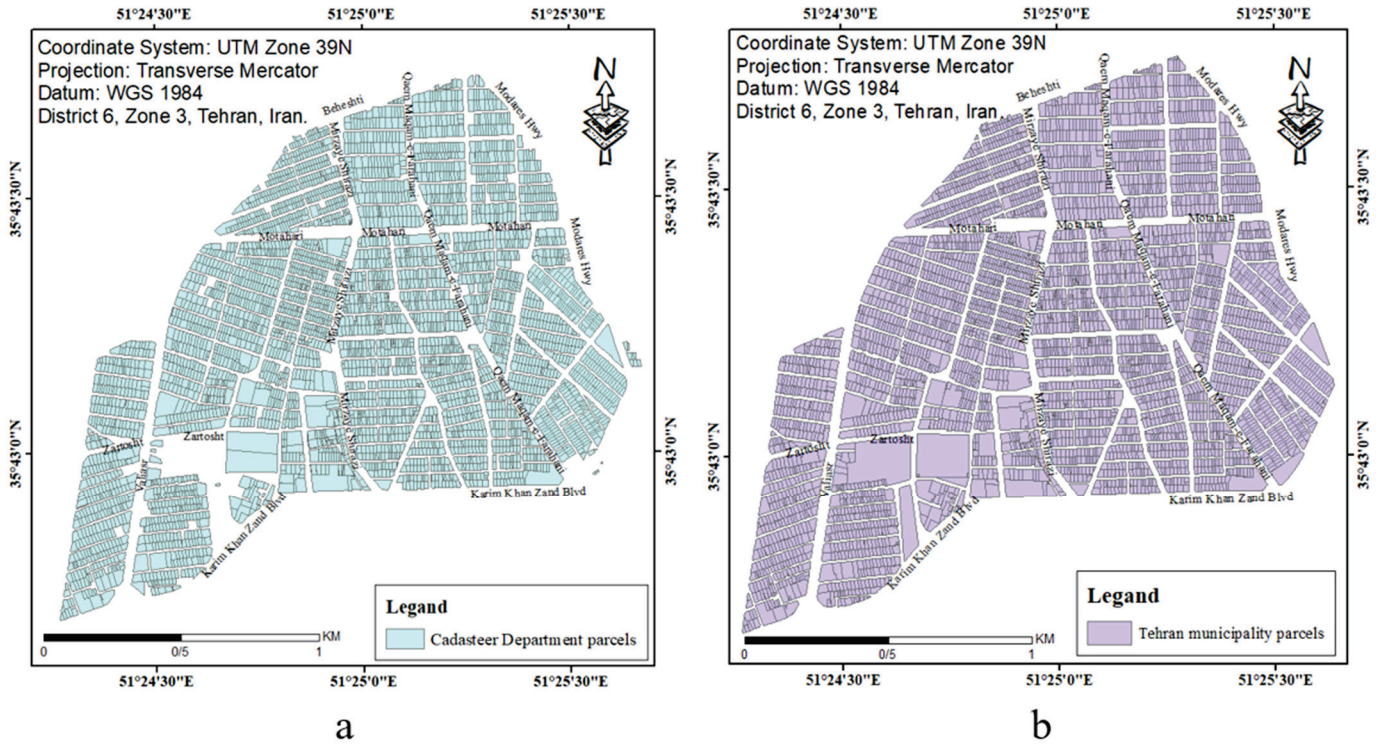


Figure 7. Employed data after preprocesses. (**a**) The Cadaster Department map. (**b**) The Tehran Municipality map.

3.4. Matching

In order to evaluate the matching, test data had to be selected from all the different matching modes. Hence, after identifying the types of parcels in the first map based on the number of vertices, five percent of each type were considered as test data to validate the matching process. A total of 230 parcels were selected for the matching validation, which was about five percent of each state randomly considered in the study area. Figure 8 shows the selected parcels for the matching validation in the Cadaster Department (base) map.

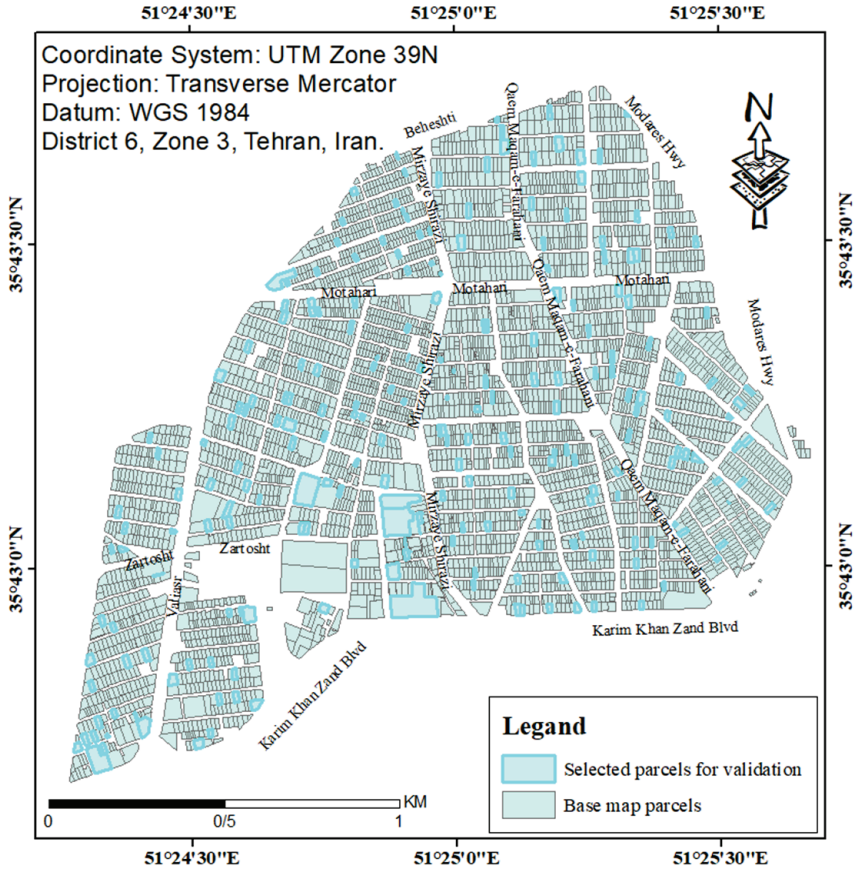


Figure 8. The selected parcels for the matching validation.

After the matching process, the elements of the confusion matrix were calculated by test data, as shown in Table 5. According to the confusion matrix, the accuracy and F1-score of matching were equal to 0.952 and 0.975, which indicates that a high matching accuracy was achieved.

Table 5. Matching validation.

	False Negative	False Positive	True Negative	True Positive
	219	0	10	1
Precision		0.956		
Recall		0.995		
F1-Score		0.975		
Accuracy		0.952		

In Table 5, confusion matrix elements have been defined as [41]:

- True Positives (TPs): correctly matched parcels;
- True Negatives (TNs): incorrectly matched parcels (the parcels were deleted in the second map while they were detected correctly by the algorithm);
- False Positives (FPs): incorrectly matched parcels;
- False Negatives (FNs): parcels that were not matched incorrectly (the parcels were not deleted in the second map but were not detected incorrectly by the algorithm).

3.5. Classification and Parcel Change Detection Modeling

Table 6 presents the parameters for tuning the models.

Table 6. Tuning parameters.

No.	Parameter	Value
1	Maximum Number of Iterations	400
2	Number of Population	400
3	Maximum Number of GA, GWO or HHO Executions per Iteration	1
4	Number of Folds in Cross Validation	5
5	Number of Iterations to Terminate the Models after Unnoticeable Changes	10
6	Value of Unnoticeable Changes to Terminate the Models	1×10^{-4}
7	GA Intersection Percentage	80
8	GA Mutation Percentage	30
9	GA Elite Percentage	Five of the best
10	GA Selection Method	Roulette wheel
11	GA Intersection and Mutation Method	Two-point Crossover

The Cadaster Department map, which was used as the base map, has 4444 parcels, and the Tehran Municipality map, which was used as a more up-to-date and richer map, has 4468 parcels. In order to train and test the models, 25% of the data, which is equal to 1110 parcels of the Cadaster Department map, were divided into two categories of training and test data. A total of 70% of these data (777 parcels) were considered as training data and 30% (333 parcels) were considered as test data. After the matching, according to Table 1 and Figures 1 and 3, some changes can be identified, including addition, deletion, integration (merging) and separation (splitting) of the parcels. For better training and testing of the models, parcels were selected from the integrated (merged), separated (split) and other parcels as training and test parcels. Table 7 shows the frequency of the different situations and the method used to select the training and test data.

Table 7. Frequencies of different matching situations and the methods used to select the number of training and test parcels from the different situations.

	Total	No. of Training Data	No. of Test Data	No. of Training and Test Data
Integrated (Merged)	323	100	33	133
Separated (Split)	85	25	10	35
Other Parcels	4036	652	290	942
Total	4444	777	333	1110

The training and test data were spatially distributed in the study area shown in Figure 9.

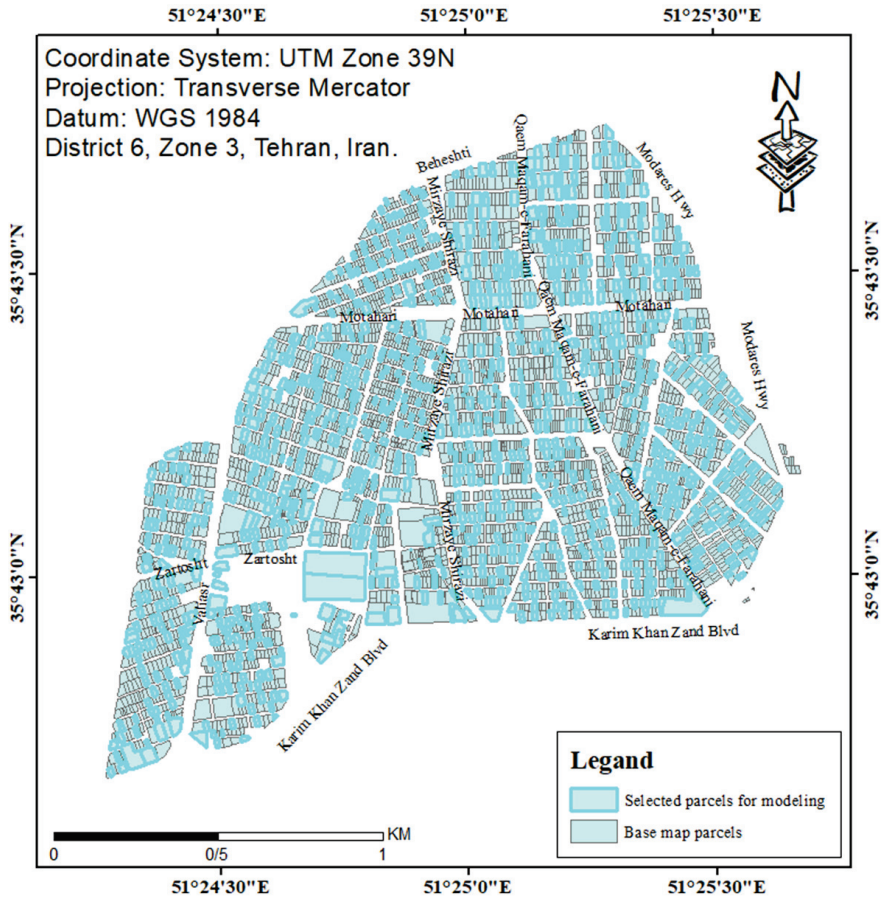


Figure 9. The parcels selected as training and test data for modeling.

To evaluate the models, in each model, the test data and outputs were used to calculate the true positives, true negatives, false positives and false negatives. Finally, precision, recall, accuracy and F1-scores were calculated for each method. Table 8 shows the comparison of the models in terms of accuracy and F1-scores for the training and test data. In Table 8, SVM-GA and RF-GA are the models with feature selection, SVM-GA-GWO and RF-GA-GWO are the models with feature selection and grey wolf optimization, and SVM-GA-HHO and RF-GA-HHO are the models with feature selection and Harris hawks optimization.

Table 8. Change detection modeling validation.

Model	TP	FN	FP	TN	Accuracy	Precision	Recall	F1-Score
SVM	746	38	237	89	0.7223	0.7589	0.9515	0.8444
SVM-GA	746	20	249	77	0.7577	0.7542	0.9745	0.8503
SVM-GA-GWO	758	26	232	94	0.7676	0.7657	0.9667	0.8546
SVM-GA-HHO	747	37	206	120	0.7811	0.7838	0.9528	0.8601
RF	722	62	129	197	0.8279	0.8484	0.9209	0.8832
RF-GA	738	46	128	198	0.8432	0.8522	0.9413	0.8945
RF-GA-GWO	744	40	122	204	0.8541	0.8591	0.9490	0.9018
RF-GA-HHO	726	58	84	242	0.8721	0.8963	0.9260	0.9109

According to Table 8, the recalls of all the models are very high and close to one, which indicates the high ability of the models to identify the changes in the parcels of the Cadaster Department base map. The RF model has a higher accuracy and F1-score than the SVM model in both the GWO and HHO optimization modes, which indicates

the greater ability of the RF model to detect changes and distinguish between changed parcels and unchanged ones compared to the SVM model. The SVM and RF models have the lowest accuracy among the models, which shows the effect of feature selection and hyperparameter optimization on improving accuracy of the models. Also, optimization by the Harris hawks optimization algorithm increased the F1-scores of the models compared to optimization by the grey wolf algorithm.

A list of features selected in each model is shown in Table 9. These features were selected by the feature selection algorithm (GA) in each model. The "N" indicates that the feature was not selected, and the "Y" indicates that the feature was selected in the desired model.

Table 9. The parameters (features) selected in the models by feature selection.

No.	Parameter	SVM	SVM-GA	SVM-GA-GWO	SVM-GA-HHO	RF	RF-GA	RF-GA-GWO	RF-GA-HHO
1	Area	Y	N	N	N	Y	N	Y	Y
2	Centroid X	Y	Y	N	Y	Y	Y	Y	N
3	Centroid Y	Y	N	N	N	Y	N	N	N
4	Middle X	Y	N	Y	N	Y	Y	Y	N
5	Middle Y	Y	Y	Y	Y	Y	N	N	N
6	Node Count	Y	Y	Y	Y	Y	Y	Y	Y
7	Minimum Extent Y	Y	Y	Y	Y	Y	Y	Y	Y
8	Maximum Extent Y	Y	N	N	N	Y	Y	Y	Y
9	Polygon Main Angle	Y	N	N	N	Y	N	N	N
10	MBB Width	Y	N	Y	N	Y	Y	Y	Y
11	MBB APodX1	Y	Y	Y	Y	Y	N	Y	N
12	MBB APodY1	Y	N	N	N	Y	N	N	N
13	MBB Orientation	Y	N	N	N	Y	Y	Y	Y
14	MBB Area	Y	N	Y	N	Y	N	N	N
15	Average Length	Y	N	N	Y	Y	N	N	Y
16	Variance Length	Y	Y	Y	N	Y	N	Y	N
17	Maximum Node	Y	N	N	N	Y	Y	Y	Y
18	Average Node	Y	N	Y	N	Y	N	Y	Y
19	Sum Node	Y	Y	Y	Y	Y	Y	N	N

According to Table 9, the features of the number of vertices (node count) of the parcels and the minimum extent Y of the parcels were selected in all the models, which indicates the importance of these features in detecting the changes in parcels with the research models. In addition, the features of centroid Y of the parcels, the polygon main angle of the parcels and MBB APodY1 were not selected in any of the models in the feature selection process, which indicates the lack of influence of these items in detecting changes in the parcels in the models. The selected features indicate the greater importance of geometric features compared to topologic and statistical features in detecting the parcel changes.

According to Table 8, the RF models with feature selection and GWO or HHO have better accuracy and F1-scores than the SVM ones. Based on the statistics of the confusion matrix in Table 8, the RF-GA-GWO model detected the changed parcels better than the RF-GA-HHO model. Hence, the RF-GA-GWO model was determined as the proposed model and implemented on the whole parcels of the base map. After training the model, other parcels that were not selected as training and test data were checked by the determined parameters and features, and the changes in the parcels were intelligently identified. Figure 10 illustrates the changed parcels detected by RF-GA-GWO. In this model, more than half of the parcels were detected as changed parcels, which indicates the existence of a

number of changes in the base map and the need for its enrichment. The parcels detected via this method were used in the next section to enrich the base map.

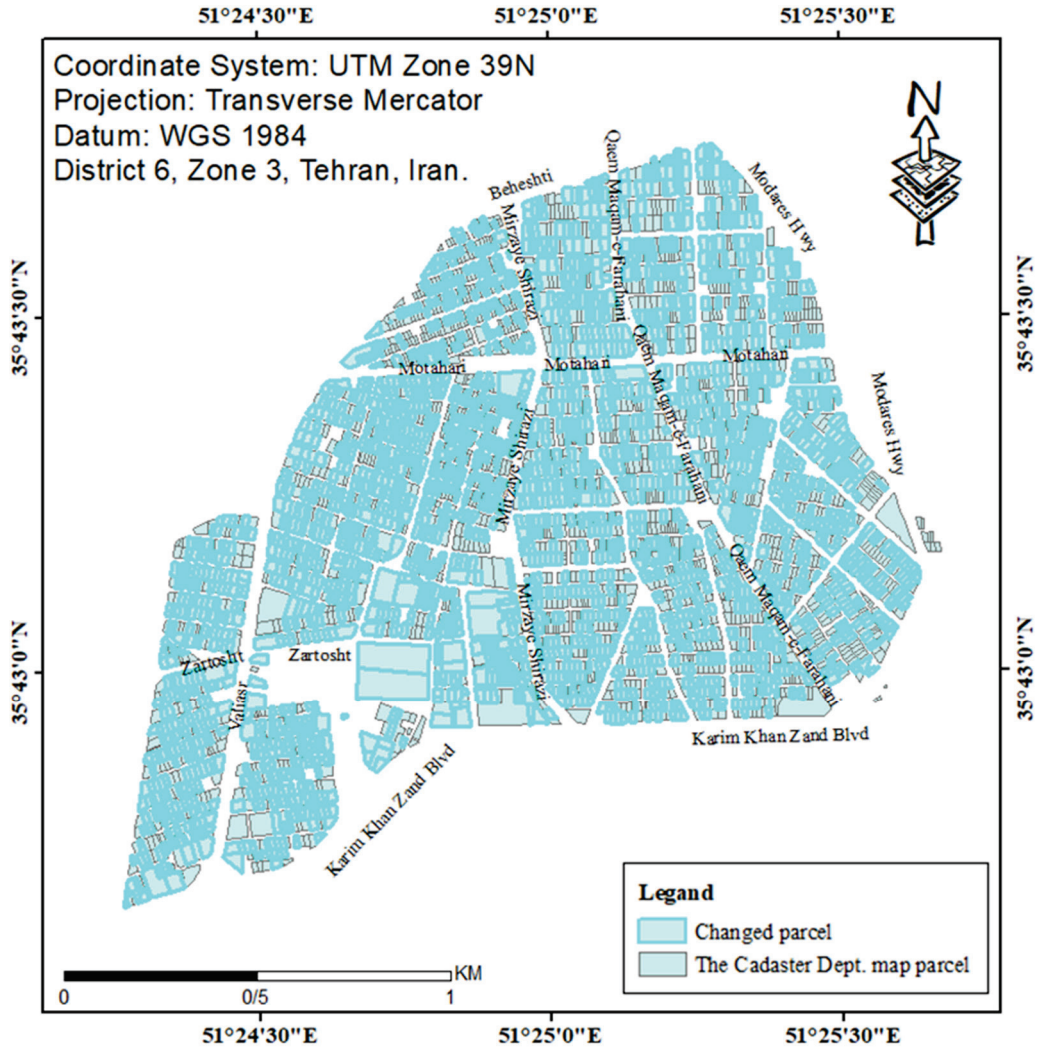


Figure 10. The changed and detected parcels in the RF-GA-GWO model.

3.6. Base Map Enrichment

After the intelligent detection of the parcels in the Cadaster Deparment map that were changed according to the Tehran Municipality map and using data mining and machine learning methods, the enrichment of the base map was performed automatically. First, the detected changed parcels in the Cadaster Department map were removed. Then, by the neighbors of the parcels in the Tehran Municipality map and the transformation parameters, the corresponding parcels in the Tehran Municipality map were located in the Cadaster Department map. The transformation and placement of the parcels from the Tehran Municipality map to the Cadaster Department map were checked by the root mean square error (RMSE). After placing all the changed parcels, to evaluate the data enrichment, the topology of the parcels was checked. In cases of topologic errors, such as the overlaying of several parcels, the incorrectly located parcels were identified and corrected. Figure 11 presents an example of the identification of the changes in the Cadaster Department map and the location of the parcels from the Tehran Municipality map.

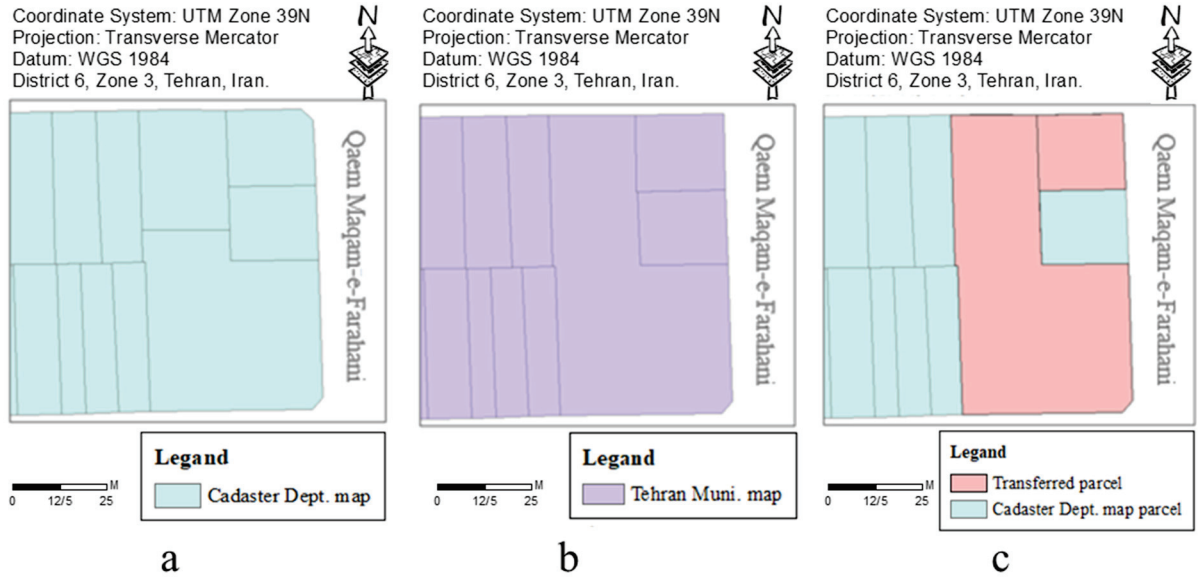


Figure 11. The intelligent change detection and map enrichment in the base map. (**a**) The Cadaster Department map before enrichment. (**b**) The Tehran Municipality map. (**c**) The Cadaster Department map after the automatic enrichment.

According to Figure 11, in the displayed area, two changes were detected in the parcels of the Cadaster Department map. The detected changes included the merging of two parcels with each other and changes in the parcel geometry. The RMSEs of the merged parcels and the geometrically modified parcels were equal to 0.00051 and 0.000113 m, respectively.

Figure 12 illustrates the intelligent detection of the separated parcels in the base map, as well as the automatic enrichment of the base map based on the transfer and placement of parcels from the Tehran Municipality map to the base map. The RMSE of the separated (split) parcel was equal to 0.000035 m.

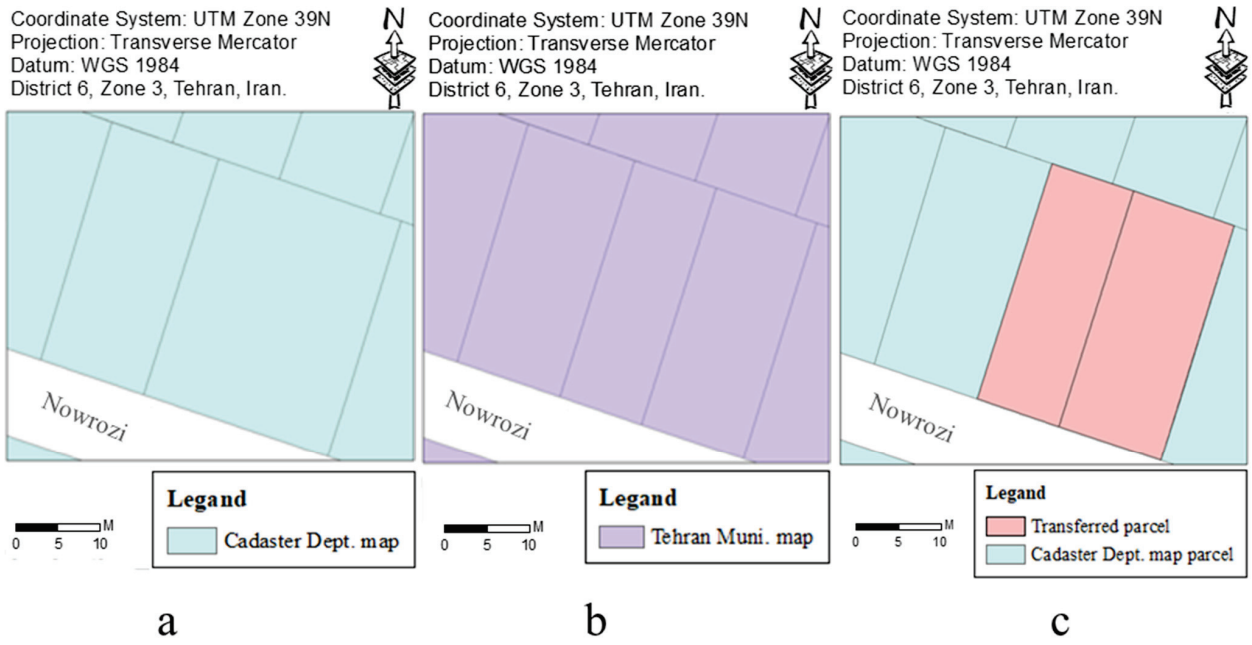


Figure 12. The intelligent change detection and map enrichment in the base map. (**a**) The Cadaster Department map before enrichment. (**b**) The Tehran Municipality map. (**c**) The Cadaster Department map after the automatic enrichment.

Figure 13 shows the topologic error (gap) caused by the removal of the changed parcel in the urban cadaster base map and its automatic enrichment. In these cases, the topologic error was resolved by correcting the parcel manually.

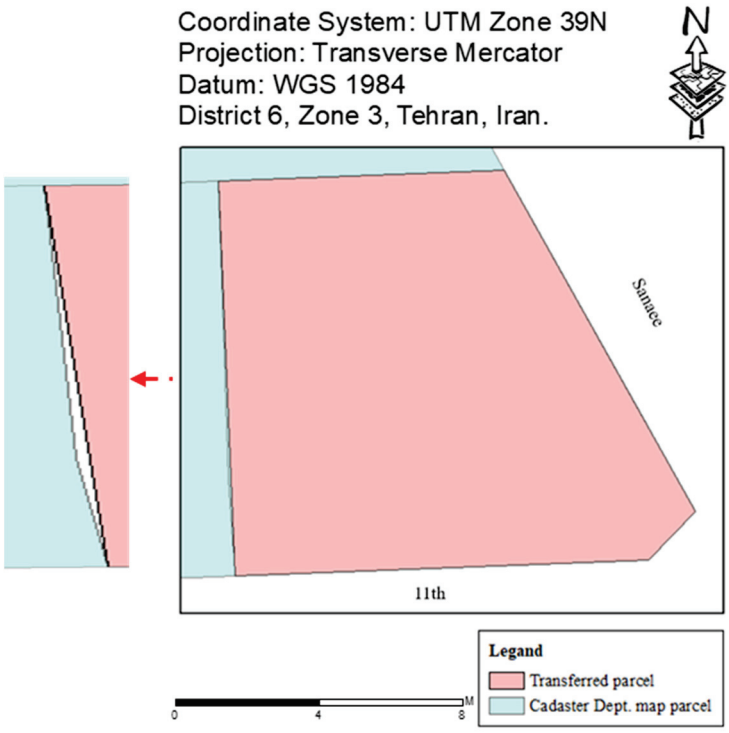


Figure 13. The topologic error (gap) in the transformation and placement of the parcel from the Tehran Municipality map to the Cadaster Department map.

4. Discussion

The matching algorithm has an accuracy of 0.952 and an F1-score of 0.975. This method was able to identify the corresponding parcels with high accuracy. On the other hand, the use of the two-way matching caused more parcels to be correctly matched, and different states, including added and split parcels, were identified correctly and with high accuracy. Also, matching by the proposed method and identifying integrated (merged) and separated (split) parcels played an important role in training and validation in the modeling. Classifying the data by matching outputs caused a good selection for the training and test data, which had a suitable and random distribution among the different states of the parcels.

Using the feature selection and optimization of hyperparameters of the models improved the detection of changed parcels in the data and its accuracy. Feature selection led to selection of the best and most suitable features in each model. Based on the high and almost similar recalls of all the models, their ability to detect changes is very high. The RF models performed better than the SVM models due to the simultaneous and parallel implementation of several basic models with each other and finally using the majority vote to determine the result. The RF-GA-GWO and RF-GA-HHO models, with accuracies of 0.8541 and 0.8721, have the highest accuracies and ability to distinguish between changed and unchanged parcels. These models, with F1-scores equal to 0.9018 and 0.9109, have the highest F1-scores and ability to detect changes and the accuracy of detecting changes among the models employed in this research. The selected features in the models showed a greater influence of geometric features in detecting the changed parcels than topologic and statistical features. According to the confusion matrix in Table 4, feature selection and hyperparameter optimization effectively increased the rate of true predictions (true positives and true negatives) and also decreased false predictions (false negatives and

false positives) in the proposed models. The RF-GA-GWO model had higher rates of true positives and false negatives, which increased the correct detection of the changed parcels. In contrast, the RF-GA-HHO model had higher rates of false positives and true negatives, which increased trust in the detected changed parcels. Hence, grey wolf optimization improved the ability of the RF algorithm to detect the changed parcels in the base map, while HHO improved the ability of the RF algorithm to detect the unchanged parcels in the base map. In general, in an area with a huge number of changed parcels, as in the base map compared to the more up-to-date map of the research study area, the RF-GA-GWO has a better performance than the RF-GA-HHO model in detecting these changes. Hence, the RF-GA-GWO model was selected as the proposed method in this research.

The automatic enrichment of the urban cadaster base map made it unnecessary to have experts check all the parcels and accelerated the enrichment of the maps. In addition, the parcels that were detected as false positives and considered as changed parcels by mistake were relocated by the proposed method. The strength of the models lay in detecting the changes in the parcels. In contrast, in detecting the unchanged parcels, they did not have the same performance as in detecting the changed parcels. Therefore, our proposed methodology more accurately covers false-positive cases in models and plays a significant role in the intelligent and automatic enrichment of maps. Automatic enrichment modified the shifting, rotation and scaling of the parcels in the Tehran Municipality map compared to those of the Cadaster Department map by affine transformation to replace the changed parcels in the base map. The existing topologic errors in the enrichment step were mostly due to the difference in the number of vertices (node counts) of the unchanged neighbors for each of the changed parcels in the two maps of Tehran Municipality and the Cadaster Department, which caused the overlapping of parcels or gaps between them. If the maps were correctly GIS readied and there were no extra vertices in the parcels, there would be fewer topologic errors in the automatic enrichment process.

Figure 14 illustrates the intelligent and automatic framework developed in this research. First, available maps and data from different organizations are collected. Then, after preprocessing, the differences, inconsistencies and errors in the maps are resolved. By calculating the center of gravity of each parcel in the maps and transferring them to other maps, the corresponding parcels are identified in the other maps. If there are new information and features for the parcels, or if there is a specific geometric shape complexity for the parcels in the investigated area, the models are retrained by the training data. Finally, the selected features and modeling parameters are extracted for the other data. If the study area is similar to the previously studied area in which the model was trained, the selected parameters and features are used in the proposed model. After modeling all the parcels and detecting the changes in the whole area, the parcels from other maps that correspond to the changed parcels in the base map are placed in the base map by the affine transformation algorithm after calculating the affine parameters from unchanged neighbor parcels for each changed parcel, then the topologic errors are removed so that the base map is intelligently and automatically enriched.

In this research, unlike previous studies [6,9,11], changes in a base map were identified by machine learning algorithms in an intelligent and automatic method, which increases the speed and accuracy of the urban map enrichment process. Moreover, relocating changed parcels automatically and updating them with the help of a more up-to-date map reduces human errors and the time required to enrich urban maps. In previous studies [6,9,11], enrichment has often been performed manually, visually and case by case, which is very time-consuming compared to the proposed model. The presented framework plays a very important role in reducing the cost and time spent by organizations in preparing urban maps or enriching urban cadastral maps by the quick and accurate use of available maps from other organizations. This framework solves some of the needs of organizations and lets them make more accurate and informed decisions.

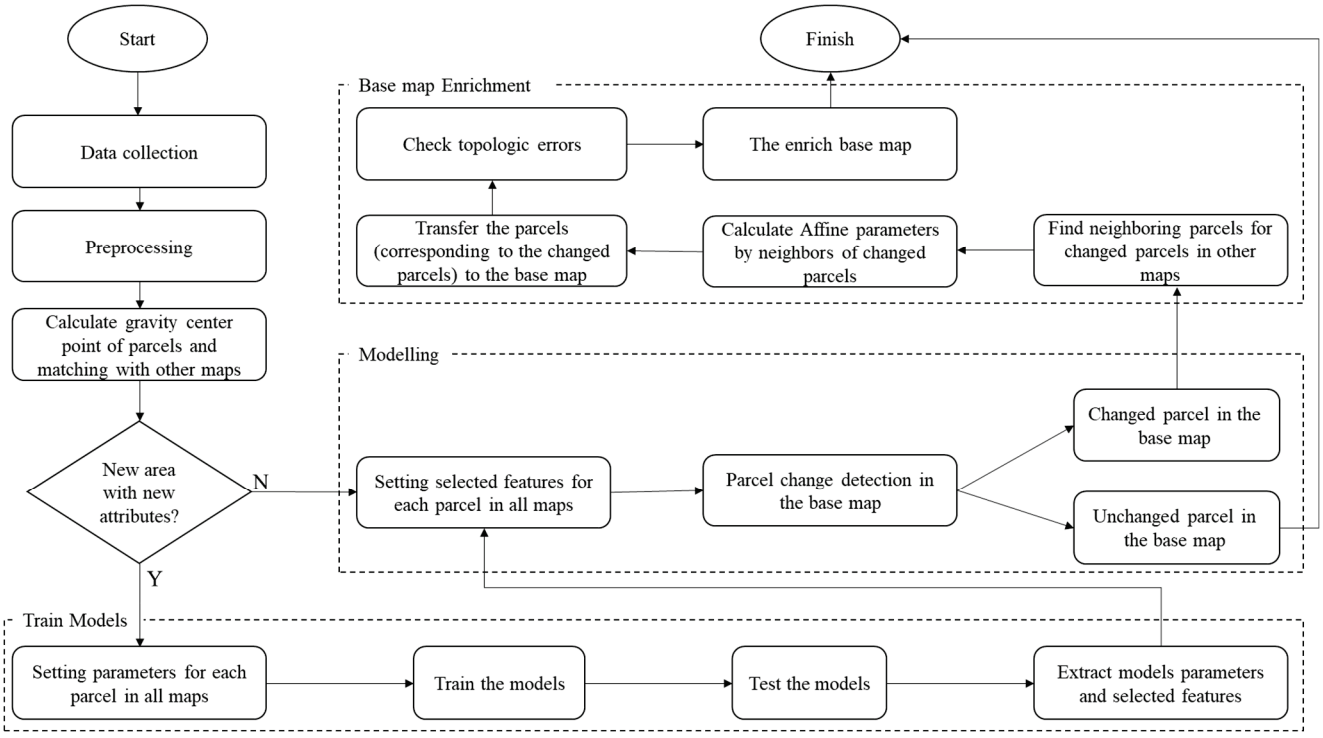


Figure 14. The proposed framework for urban base map enrichment.

Since the juridical map and registered parcels are fixed properties and obtained from the special administrative procedures, these parcels cannot be changed and replaced by other parcels without legal processing. In this research, the base map from the Cadaster Department was produced by the NCC and had no registered parcel. This base map is usually used beside another dataset for registered parcels to handle new requests for property registration. Hence, enriching this base map reduces the Cadastral Department process for registration. On the other hand, other organizations, such as utilities and infrastructure organizations, can effectively use the proposed framework to enrich their maps for better decision making and urban management.

5. Conclusions

The preparation of accurate cadastral maps in urban areas is a time-consuming and costly process for urban managent organizations. In addition, the enrichment and upgrading of maps by other organizations' maps needs huge manpower, time and funding. Hence, intelligent and automatic enrichment of an existing map by the maps of other organizations and more up-to-date maps plays a vital role in informed urban management, enabling timely decisions with reduced costs. Since parcels are an important part of urban maps, an intelligent and automatic framework is essential to enrich two-dimensional maps of parcels in urban areas with urban maps of other organizations in the metropolises and cities of a country. This research focused on enrichment of the parcels in the Cadaster Department map from a geometric point of view and by the most up-to-date maps available in Tehran Municipality. After the data preprocessing, matching was performed by each parcel gravity center point and checking the position of the points in the other map. The matching process was implemented in both of the maps to find all types of matching and relations. To validate the process, five percent of the base map parcels were selected randomly based on the number of vertices of parcels. The accuracy and F1-scores were 0.952 and 0.975 for the matching algorithm. Based on different matching relations and different changes in the base map, 25 percent of the whole data (1110 parcels from 4444 parcels) were considered as training and test data. The training data included 70 percent of these data, and 30 percent of the data were considered as the test data. A total of 38 different geometric, topologic

and statistical parameters were calculated for each parcel in both of the maps. After correlation matrix calculation, 19 parameters were selected for the modeling. Classification of the changed and unchanged parcels was implemented by SVM and RF algorithms. To improve the performance and accuracy of the models, feature selection by the GA and hyperparameter optimization by the GWO and HHO algorithms was performed in each model, which improved the accuracy of the models by about five percent and the F1-scores by about two percent. The proposed models were RF-GA-GWO and RF-GA-HHO. The RF-GA-GWO improved the model for detecting the changed parcels in the base map, and the RF-GA-HHO improved the model for detecting unchanged parcels in the base map. After training and validating the models, all data of the base map were classified into changed and unchanged parcels by the proposed models. The RF-GA-GWO model was selected for the base map enrichment phase, with more than 50 percent changed parcels in the modeling phase. In the base map enrichment step, the changed parcels were deleted from the base map and replaced with the corresponding parcels from the more up-to-date map automatically by neighbors of the changed parcels in both of the maps and affine transformation. Each parcel was evaluated by the RMSE of affine transformation and topology checking undertaken after the base map enrichment. Finally, an intelligent and automatic framework was proposed for map enrichment with a more up-to-date map by the machine learning algorithms.

Data access limitation was one of the main issues of this research. More information about parcels, such as owner's names and land use. in each map may help to match the parcels better and cover other parts of map enrichment, such as semantics. Parcels with holes or parcels with their gravity centers located outside of their corresponding parcels in the other map may have caused errors in the matching process. There were none of these types of parcels in the study area; however, these parcels can be detected by point-in-polygon analysis, and the matching process can be corrected before the modeling. Since the accuracy and selected features in both of the models were close, the ensemble methods for integrating the models' outputs were not very useful. Other models that cover other parts of feature space can be integrated with these models by ensemble algorithms, such as voting and stacking, to cover uncertainty [32].

In future research, we are going to consider other types of data in the maps to enrich the base map. In addition to the geometric point of view, other data like descriptive information can be considered. Another suggestion for future research is to use more complex transformations, such as projective transformation, to evaluate the accuracy of the enrichment phase. The outputs of the models can be integrated with each other by different algorithms, such as stacking and voting [32].

Author Contributions: Conceptualization, Alireza Hajiheidari, Mahmoud Reza Delavar and Abbas Rajabifard; methodology, Alireza Hajiheidari; software, Alireza Hajiheidari; validation, Alireza Hajiheidari; formal analysis, Alireza Hajiheidari; investigation, Alireza Hajiheidari; resources, Alireza Hajiheidari and Mahmoud Reza Delavar; data curation, Alireza Hajiheidari; writing—original draft preparation, Alireza Hajiheidari; writing—review and editing. Mahmoud Reza Delavar and Abbas Rajabifard; visualization, Alireza Hajiheidari; supervision, Mahmoud Reza Delavar and Abbas Rajabifard; project administration, Mahmoud Reza Delavar. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Molinero-Parejo, R.; Aguilera-Benavente, F.; Gómez-Delgado, M.; Shurupov, N. Combining a Land Parcel Cellular Automata (LP-CA) Model with Participatory Approaches in the Simulation of Disruptive Future Scenarios of Urban Land Use Change. *Comput. Environ. Urban Syst.* **2023**, *99*, 101895. [CrossRef]

2. Hajji, R.; El Asri, H.; Ez-Zriouli, C. Upgrading to 3D Cadastre in Morocco: Lessons Learned from Benchmarking of International 3D Cadastral Systems. *Land Use Policy* **2023**, *128*, 106605. [CrossRef]
3. Govedarica, M.; Radulović, A.; Sladić, D. Designing and Implementing a LADM-Based Cadastral Information System in Serbia, Montenegro and Republic of Srpska. *Land Use Policy* **2021**, *109*, 105732. [CrossRef]
4. Tepe, E.; Safikhani, A. Spatio-Temporal Modeling of Parcel-Level Land-Use Changes Using Machine Learning Methods. *Sustain. Cities Soc.* **2023**, *90*, 104390. [CrossRef]
5. Indrajit, A.; van Loenen, B.; Ploeger, H.; van Oosterom, P. Developing a Spatial Planning Information Package in ISO 19152 Land Administration Domain Model. *Land Use Policy* **2020**, *98*, 104111. [CrossRef]
6. Bacior, S. Austrian Cadastre Still in Use—Example Proceedings to Determine the Legal Status of Land Property in Southern Poland. *Land Use Policy* **2023**, *131*, 106740. [CrossRef]
7. Hajiheidari, A.R.; Delavar, M.R.; Rajabifard, A. Cadastral and urban maps enrichments using smart spatial data fusion. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *10*, 263–269. [CrossRef]
8. Cienciała, A.; Sobolewska-Mikulska, K.; Sobura, S. Credibility of the Cadastral Data on Land Use and the Methodology for Their Verification and Update. *Land Use Policy* **2021**, *102*, 105204. [CrossRef]
9. Ercan, O. Evolution of the Cadastre Renewal Understanding in Türkiye: A Fit-for-Purpose Renewal Model Proposal. *Land Use Policy* **2023**, *131*, 106755. [CrossRef]
10. Negri, R.G.; Frery, A.C. A General and Extensible Framework for Assessing Change Detection Techniques. *Comput. Geosci.* **2023**, *178*, 105390. [CrossRef]
11. Čeh, M.; Gielsdorf, F.; Trobec, B.; Krivic, M.; Lisec, A. Improving the Positional Accuracy of Traditional Cadastral Index Maps with Membrane Adjustment in Slovenia. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 338. [CrossRef]
12. Safra, E.; Kanza, Y.; Sagiv, Y.; Doytsher, Y. Integrating Data from Maps on the World-Wide Web. In Proceedings of the International Symposium on Web and Wireless Geographical Information Systems, Hong Kong, China, 4–5 December 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 180–191.
13. Pullar, D.; Donaldson, S. Accuracy Issues for Spatial Update of Digital Cadastral Maps. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 221. [CrossRef]
14. Song, A.-R.; Park, S.; Kim, Y.-I. *Updating Cadastral Maps Using Deep Convolutional Networks and Hyperspectral Imaging*; Daejeon Convention Center(DCC): Daejeon, Korea, 2019.
15. Wierzbicki, D.; Matuk, O.; Bielecka, E. Polish Cadastre Modernization with Remotely Extracted Buildings from High-Resolution Aerial Orthoimagery and Airborne LiDAR. *Remote Sens.* **2021**, *13*, 611. [CrossRef]
16. Ali, Z.; Tuladhar, A.; Zevenbergen, J. An Integrated Approach for Updating Cadastral Maps in Pakistan Using Satellite Remote Sensing Data. *Int. J. Appl. Earth Obs. Geoinf.* **2012**, *18*, 386–398. [CrossRef]
17. Puniach, E.; Bieda, A.; Ćwiąkała, P.; Kwartnik-Pruc, A.; Parzych, P. Use of Unmanned Aerial Vehicles (UAVs) for Updating Farmland Cadastral Data in Areas Subject to Landslides. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 331. [CrossRef]
18. Wang, S.; Li, W. GeoAI in Terrain Analysis: Enabling Multi-Source Deep Learning and Data Fusion for Natural Feature Detection. *Comput. Environ. Urban Syst.* **2021**, *90*, 101715. [CrossRef]
19. Girard, N.; Charpiat, G.; Tarabalka, Y. Aligning and Updating Cadaster Maps with Aerial Images by Multi-Task, Multi-Resolution Deep Learning. In Proceedings of the 14th Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; pp. 675–690, ISBN 978-3-030-20872-1.
20. Fetai, B.; Grigillo, D.; Lisec, A. Revising Cadastral Data on Land Boundaries Using Deep Learning in Image-Based Mapping. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 298. [CrossRef]
21. Nyandwi, E.; Koeva, M.; Kohli, D.; Bennett, R. Comparing Human Versus Machine-Driven Cadastral Boundary Feature Extraction. *Remote Sens.* **2019**, *11*, 1662. [CrossRef]
22. Samal, A.; Seth, S.; Cueto, K. A Feature-Based Approach to Conflation of Geospatial Sources. *Int. J. Geogr. Inf. Sci.* **2004**, *18*, 459–489. [CrossRef]
23. Kim, J.O.; Yu, K.; Heo, J.; Lee, W.H. A New Method for Matching Objects in Two Different Geospatial Datasets Based on the Geographic Context. *Comput. Geosci.* **2010**, *36*, 1115–1122. [CrossRef]
24. Wang, Y.; Lv, H.; Chen, X.; Du, Q. A PSO-Neural Network-Based Feature Matching Approach in Data Integration. In *Cartography-Maps Connecting the World*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 189–219. [CrossRef]
25. Nürnberg, R. Calculating the Area and Centroid of a Polygon in 2d. 2013. Available online: http://wwwf.imperial.ac.uk/rn/centroid.pdf (accessed on 25 January 2024).
26. Zhang, X.; Ai, T.; Stoter, J.; Zhao, X. Data Matching of Building Polygons at Multiple Map Scales Improved by Contextual Information and Relaxation. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 147–163. [CrossRef]
27. Xavier, E.M.; Ariza-López, F.J.; Urena-Camara, M.A. A Survey of Measures and Methods for Matching Geospatial Vector Datasets. *ACM Comput. Surv.* **2016**, *49*, 1–34. [CrossRef]
28. Awad, M.; Khanna, R. Support Vector Regression. In *Efficient Learning Machines*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 67–80.
29. Wu, X.; Kumar, V.; Ross Quinlan, J.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.J.; Ng, A.; Liu, B.; Yu, P.S. Top 10 Algorithms in Data Mining. *Knowl. Inf. Syst.* **2008**, *14*, 1–37. [CrossRef]
30. Smola, A.J.; Schölkopf, B. A Tutorial on Support Vector Regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]

31. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann: Burlington, MA, USA, 2012.
32. Zhang, C.; Ma, Y. *Ensemble Machine Learning: Methods and Applications*; Springer: Berlin/Heidelberg, Germany, 2012; ISBN 1-4419-9325-8.
33. Kuncheva, L.I. *Combining Pattern Classifiers: Methods and Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2014; ISBN 1-118-31523-5.
34. Liaw, A.; Wiener, M. Classification and Regression by randomForest. *R News* **2002**, *2*, 18–22.
35. Mirjalili, S.; Dong, J.S.; Lewis, A. *Nature-Inspired Optimizers*; Springer: Cham, Switzerland, 2020; pp. 69–85.
36. Haupt, R.L.; Haupt, S.E. *Practical Genetic Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2004; ISBN 0-471-67175-4.
37. Srinivas, M.; Patnaik, L.M. Genetic Algorithms: A Survey. *Computer* **1994**, *27*, 17–26. [CrossRef]
38. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
39. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris Hawks Optimization: Algorithm and Applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
40. Chai, T.; Draxler, R.R. Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)?—Arguments against Avoiding RMSE in the Literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [CrossRef]
41. Chicco, D.; Jurman, G. The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation. *BMC Genom.* **2020**, *21*, 6. [CrossRef]
42. Powers, D.M. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. *arXiv* **2020**, arXiv:2010.1606.
43. Gray, A.; Abbena, E.; Salamon, S. Modern Differential Geometry of Curves and Surfaces with Mathematica. 2006. Available online: https://hdl.handle.net/2318/104908 (accessed on 25 January 2024).
44. Tehran Municipality. *Tehran Statistical Yearbook 2021–2022*; Information and Communication Technology Organization of Tehran Municipality: Theran, Iran, 2022.
45. Hajiheidari, A.; Delavar, M.R.; Rajabifard, A. *Challenges of Integrating Cadastral Map and Urban Data Infrastructure in Iran*, 4th ed.; International Geoinformation Days (IGD): Tabriz, Iran, 2022; pp. 155–158.
46. Mohammadi, H. *The Integration of Multi-Source Spatial Datasets in the Context of SDI Initiatives*; University of Melbourne: Melbourne, Australia; Centre for Spatial Data Infrastructures and Land Administration: Carlton, Australia, 2008.
47. Song, X.; Cheng, C.; Zhou, C.; Zhu, D. Gestalt-Based Douglas-Peucker Algorithm to Keep Shape Similarity and Area Consistency of Polygons. *Sens. Lett.* **2013**, *11*, 1015–1021. [CrossRef]
48. National Cartographic Center of Iran. *Standards and Instructions for Public Display and Cartography*; National Cartographic Center of Iran: Tehran, Iran, 2012.
49. National Cartographic Center of Iran. *Multi-Scale Spatial Data Model*; National Cartographic Center of Iran: Tehran, Iran, 2021.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

Spatial Process Analysis of the Evolution of Farmland Landscape in China

Yan Fu [1,2], Qingwen Qi [1,2], Lili Jiang [1,2,*] and Yapeng Zhao [1,2]

[1] State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China; fuy.18b@igsnrr.ac.cn (Y.F.); qiqw@igsnrr.ac.cn (Q.Q.); zhaoyp.18b@igsnrr.ac.cn (Y.Z.)
[2] College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: jiangll@igsnrr.ac.cn; Tel.: +86-18810615373

Abstract: Accurately identifying the patterns of evolution in farmland plays an important role in optimizing farmland management. The aim of this study is to classify the evolution patterns of farmland in China and explore related mechanisms, providing a reference for constructing a systematic farmland management plan. Using land cover data from five periods in China, nine types of farmland evolution process are described and identified based on landscape process models. We analyzed these processes' spatiotemporal dynamics and, by examining regional variations, achieved a zoned mapping of China's farmland evolution. In this study, we combined natural and socioeconomic factors to analyze the mechanisms driving the evolution of farmland landscapes in China. The results indicated that from 1980 to 2020, areas of both lost and restored farmland showed a trend of first increasing and then decreasing, while the total area of farmland fluctuated. The remaining farmland types consisted mainly of core and edge. Their distribution was similar to that of the major agricultural regions in China. Expansion was the main means of farmland restoration. Farmland fragmentation was widespread, and, over time, it became increasingly severe. Shrinkage and subdivision dominated the farmland fragmentation. Altitude and slope had the greatest impact on the evolution patterns of farmland. Increasing urban industrialization and an increase in population density led to an increase in the demand for food production, which placed greater demands on the farmlands in the region. The farmland evolution pattern is a result of the interactions among multiple factors.

Keywords: evolution process of farmland; farmland fragmentation; farmland restoration

1. Introduction

Farmlands are a fundamental unit of agricultural production and important for ensuring food security and maintaining societal stability [1–3]. Farmland landscape patterns show the ecological status of and spatial variations in farmland within a region [4,5]. Several studies from Europe, Asia and Africa have shown that the evolution of farmland landscape patterns has a profound impact on farmland biodiversity, ecosystem services, human–wildlife conflicts and food security [6–12]. A loss of farmland exacerbates the fragmentation of farmland landscapes, leading to a decrease in agricultural production potential and threats to food production security [13,14]. The systematic and effective restoration of farmland can increase the scale of agricultural production and alleviate pressures on food production [15,16]. Quantitatively analyzing the dynamic evolution patterns of farmland landscapes helps decision makers understand the process of farmland resource development and formulate reasonable farmland management strategies [5].

Research on the evolution of farmland landscape patterns is mostly carried out for specific static time nodes and uses landscape indices, comprehensive landscape evaluation models and other methods to explore the distribution characteristics of farmland landscapes

at the national, provincial, municipal and county levels [17–23]. This type of research cannot provide a systematic description of the dynamic patterns in farmland evolution between time nodes; therefore, it cannot show the patterns of farmland landscape evolution. The theory of the landscape process in landscape ecology can make up for the deficiencies mentioned above. A landscape process model was validated in quantifying the spatial evolution of landscapes in the United States, Nepal, China and other countries [24–28]. Based on topology theory, landscape fragmentation models, landscape restoration process models and other models have been used to classify the evolution of landscapes over a period of time. The classification results reflect the specific patterns of landscape evolution during a period and have clear spatial implications.

Spatial analysis is an important means to study geospatial phenomena. It can realize the visualization of geospatial data and identify patterns and spatial relationships that are not easy to find using traditional data analysis methods. Geospatial analysis methods, such as the gravity center model, superposition analysis, correlation analysis and geoscience information Tupu, have been applied and verified in the fields of farmland environment, farmland quality evaluation, farmland preservation policy and land use transitions [6,29–32]. Farmland landscapes may have different evolution patterns and development trends in different periods and regions. The advantages of geospatial methods help them to mine spatial information related to farmland landscape evolution and statistically analyze the spatial law of evolution.

China's challenge of balancing a large population with limited land resources underscores the value of agricultural landscapes as essential assets. The current understanding of the spatiotemporal trends and drivers behind China's farmland landscape evolution remains inadequate. By leveraging multi-temporal farmland cover data and employing landscape process models across various scales and periods, this study aims to dissect the evolution of China's farmland landscape. It endeavors to map the zoning of farmland evolution from diverse angles to illuminate regional discrepancies. Through the analysis of natural and socioeconomic factors in correlation with different evolutionary processes, this research seeks to determine the mechanisms driving farmland changes, offering insights to refine farmland management practices.

2. Materials and Methods

2.1. Data Source and Preprocessing

The data used in this article include land cover data, environmental data, such as elevation, annual temperature and annual precipitation, and social and economic statistical data for Chinese cities.

The land cover maps for 1980, 1990, 2000, 2010 and 2020 were obtained from the Resource and Environment Data Cloud Platform (http://www.resdc.cn (accessed on 1 December 2023)). The dataset was generated by human visual interpretation on Landsat TM images from the United States Landsat, with a spatial resolution of 1000 m. The maps include 6 primary types and 25 secondary types of farmland, forestland, grassland, water area, construction land and unused land. To provide a basis for subsequent data analysis, the 25 subcategories of the land cover map were first reclassified into two categories: farmland and non-farmland. The sources of environmental data and socioeconomic statistical data are shown in Table 1.

Table 1. Description of the factors.

	Factors	Calculation	Data Source
Natural factors	Mean annual temperature	-	Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences.
	Mean annual precipitation	-	
	Mean annual accumulated temperature	-	
	Elevation	-	
	Slop	-	

Table 1. *Cont.*

	Factors	Calculation	Data Source
Socioeconomic factors	Mean population density	-	Provincial Statistical Yearbook
	Industrialization level	Added value of secondary industry/gross national product	
Technology factor	Fertilizer utilization level	Total agricultural output value/fertilizer input	
	Agricultural mechanization level	Total machinery power/agricultural practitioners	Provincial Statistical Yearbook
	Mean education level of the people		

2.2. Assessment of the Spatial Process of Farmland Change

During the evolution process, farmland change is generally categorized into three states: lost farmland, gained farmland and remaining farmland [25]. The types of lost farmland include perforation, subdivision, shrinkage and attrition. Perforation means that a lost patch is adjacent to only one remaining patch. Subdivision means that a lost patch is adjacent to at least two remaining patches and adjacent to a non-farmland patch. Shrinkage means that a lost patch is adjacent to only one remaining patch and is also adjacent to a non-farmland patch. Attrition means that a lost patch is only adjacent to a non-farmland patch. These types can cause the fragmentation of farmland and are not conducive to large-scale agricultural production. The types of gained farmland include increment and expansion. Expansion means that a gained patch is adjacent to a remaining patch. Increment means that a gained patch is surrounded by patches of non-farmland. These types are effective measures for farmland conservation. Expansion can significantly increase the connectivity among patches of farmland. Increments can ensure the quantity and quality of stable farmland resources to a certain extent. The remaining farmland can be of three types: core, edge and scattered areas. Core means that the remaining farmland is adjacent only to the remaining farmland grid. The core provides suitable conditions for large-scale production. Edge refers to a grid of remaining farmland adjacent to both core and non-farmland. It is a transition zone between agricultural production and non-agricultural activities, which can effectively buffer the impact of non-agricultural production on agricultural production. The concept of scattered areas refers to the remaining farmland grids that are adjacent only to non-farmland. The topological relationships and definitions of the evolution patterns of farmland are shown in Figure 1. Green represents remaining farmland patches, blue represents lost farmland patches, orange represents gained farmland patches and gray represents non-farmland patches.

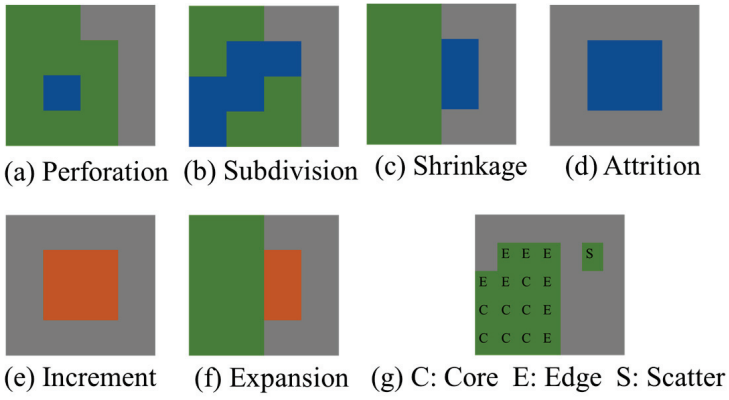


Figure 1. (**a**–**d**) Four types of lost farmland. (**e**,**f**) Two types of gained farmland. (**g**) Three types of remaining farmland.

The spatial process of farmland change was realized using ArcMap 10.5 software. The process is shown in Figure 2. Land cover map I and land cover map II represent two datasets at different time points. By overlaying the two farmland maps, we identified the lost farmland (LF), the remaining farmland (RF), the gained farmland (GF) and the non-farmland (NF). Then, using the regional group tool, we merged the four types of farmland pixels into lost farmland patches (LFPs), remaining farmland patches (RFPs), gaining

farmland patches (GFPs) and non-farmland patches (NFPs). Each patch had a unique ID. With the neighborhood analysis tool, we further obtained the adjacency relationships between the LFP and the RFP, between the LFP and NFP, between the GFP and RFP, and between the GFP and the NFP. Based on the classification principle, the final classification results for farmland change were obtained. In this way, we generated a series of maps showing farmland fragmentation and restoration processes.

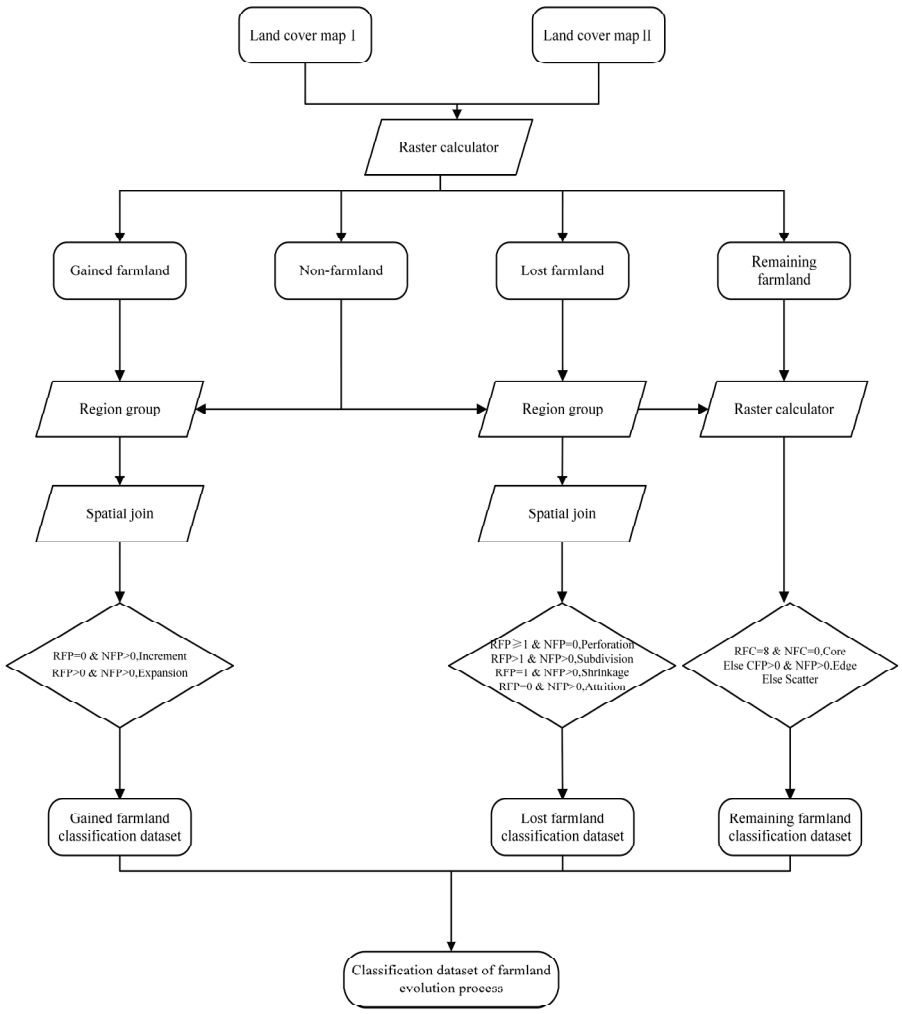


Figure 2. The process for obtaining spatial changes in farmland.

2.3. Multiangle Zoning of Farmland Landscapes

To better characterize the regional differences in the evolution of farmland landscapes and to provide a decision-making tool for farmland management, in this study, we dynamically divided the farmland landscapes into remaining farmland composition, gained farmland composition and lost farmland. In China, a county is the smallest unit that can be analyzed to regulate agricultural production and formulate farmland management plans, so the zoning units were based on counties. The scale advantage index was calculated for nine types of cultivated land [33]. The formula is as follows:

$$D_{ij} = \frac{FA_{ij}/FA_{iJ}}{FA_j/FA_J} \quad (1)$$

where D_{ij} is the scale advantage index of farmland type j in county i; FA_{ij} is the area of farmland type j in county i; FA_{iJ} is the total area of farmland type J in county i; FA_j is the total area of farmland type j in China; and FA_J is the total area of farmland type J in the country.

The specific zoning principles are as follows:

(I) Considering the impacts of core, edge and scattered-type landscapes on agricultural production, we divided agricultural production zoning into hierarchical levels. First, counties with a core scale index greater than 1 were classified as agricultural core areas (ACAs). Second, counties with an edge scale index greater than 1 were defined as agricultural protected areas (APAs). Then, counties with a scattered scale index greater than 1 were defined as agricultural improvement areas (AIAs). Finally, counties in which all three scale indices were less than 1 were defined as general agricultural areas (GAAs).

(II) Considering the means of farmland restoration, counties with an increase in the advantage index greater than 1 were defined as increment advantage areas (IAAs). Counties with an expansion advantage index greater than 1 were defined as expansion advantage areas (EAAs). Counties in which both advantage indices were less than 1 were defined as general restoration areas (GRAs).

(III) The more types of lost farmland there are, the more complex the fragmentation process of farmland in the county. If the advantage index of three or more types of lost farmland in a county was greater than 1, the county was defined as an area of extremely complex fragmentation (ECAF). If the advantage index of two types of lost farmland in the county was greater than 1, the county was defined as an area of complex fragmentation (CAF). If there was only one type of farmland loss with an advantage index greater than 1 in a county, it was defined as a fragmented area (AF). If the four lost farmland advantage indices were less than 1, then the county was classified as a generally fragmented area (GAF).

2.4. Correlation Analysis

To further explore the process of spatial changes of farmland, socioeconomic statistics and environmental factors were selected for correlation analysis. Prefecture-level cities are the basic unit for specifying agricultural policies. Considering the availability of data, we chose prefecture-level cities as the analysis unit. The administrative boundaries of the prefecture-level cities were used to extract the proportions of the nine farmland change processes in each city. We calculated the Spearman correlation coefficient between the percentages of different farmland changes and the impact factors. The greater the absolute value of the Spearman coefficient, the stronger the correlation between two factors, and the sign of the value indicates the direction of the correlation [34]. The calculation formula is as follows:

$$R_s = 1 - \frac{6\sum d^2}{n(n^2-1)} \quad (2)$$

where d is the rank difference in each pair of observations after XY is deranked, and n is the number of all observation pairs.

3. Results

3.1. Multiscale Analysis of the Dynamic Evolution of Farmland

3.1.1. Country Level

Using the farmland reclassification data from the fifth period, we obtained a process map for spatial change and a statistical map of farmland landscape, as shown in Figures 3 and 4, respectively. During the four periods, the quantity of remaining farmland in China was relatively stable, at 1,750,000 square kilometers. The remaining farmland was mainly marginal farmland. The amount of gained farmland showed a trend of first increasing and then decreasing. The greatest amount of gained farmland was present between 1999 and 2000, at 64,000 square kilometers. Farmland expansion was the main cause of farmland gain. Although the proportion has gradually decreased, it still exceeds 75%. The area of farmland loss remained relatively high for a long period of time. From 1990 to 2010, China lost approximately 70,000 square kilometers of farmland resources. After 2010, the area of loss decreased. Farmland loss occurred in the four periods mainly due to shrinkage and subdivision. During the period of 2000–2010, the proportions of shrinkage and subdivision

reached 46% and 43%, respectively. Figure 4 shows the center of gravity distribution of various types of farmland. The center of gravity movement path for expansion and increment was "northwest—south—north". The center of gravity movement path for the core areas was distributed in the North China Plain, and the scatter was distributed in the northern region of Hubei Province. The center of gravity of edges underwent a significant change, with a path of "northwest—southwest—northeast". The four types of LF have similar paths of center of gravity movement, which were "southwest—south—northeast".

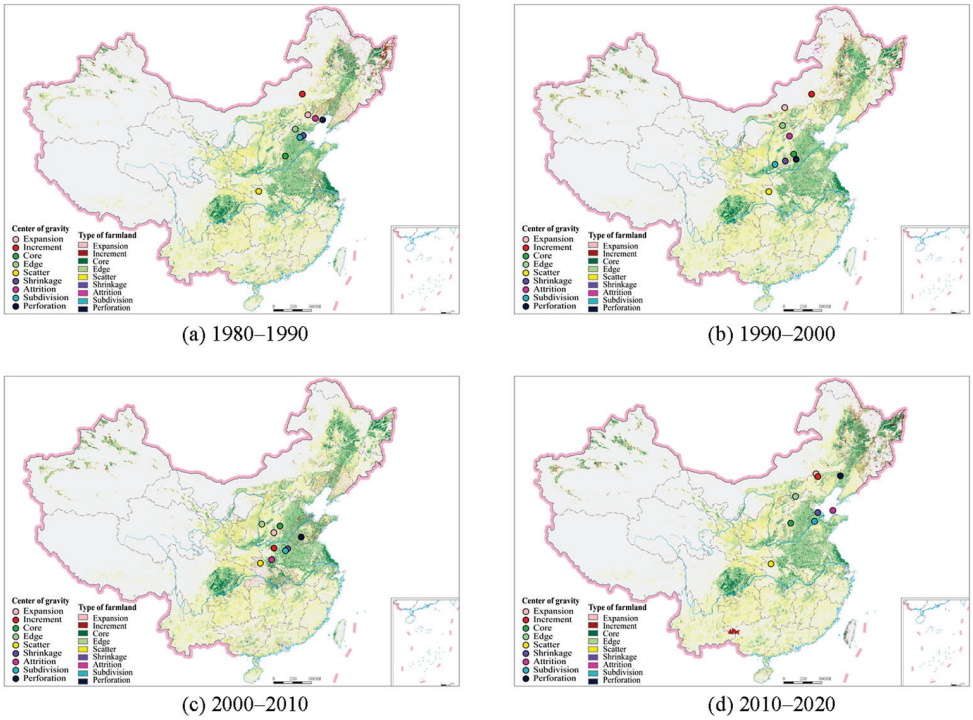


Figure 3. Spatial and temporal distribution of the processes of spatial changes of farmland over four periods.

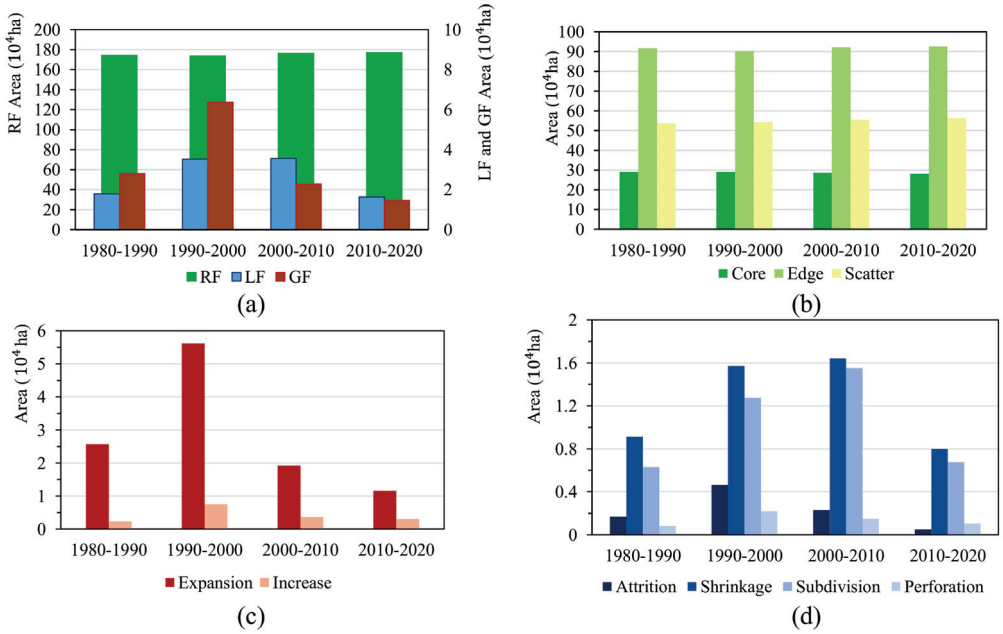
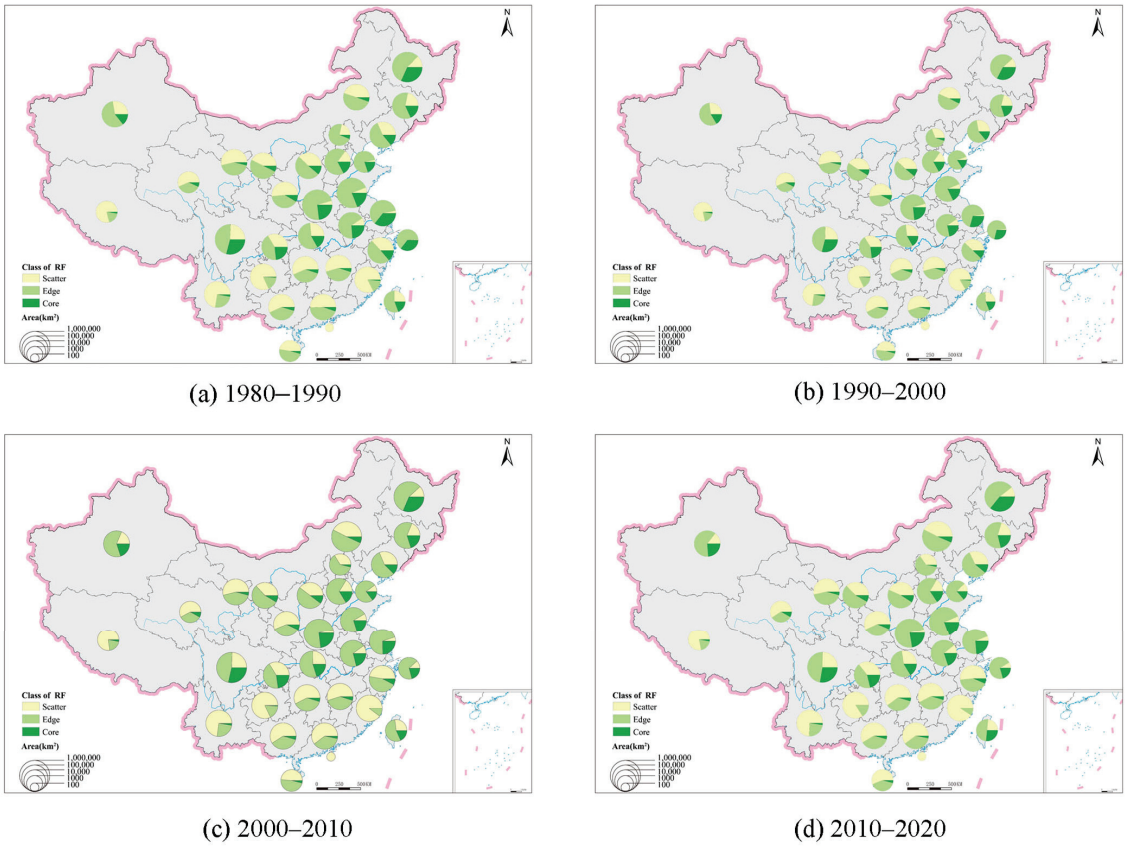


Figure 4. (**a**) Areas of RF, LF and GF. (**b**) Areas of core, edge and scatter. (**c**) Areas of expansion and increment. (**d**) Areas of attrition, shrinkage, subdivision and perforation.

3.1.2. Provincial Level

Figure 5 shows the composition of various types of farmland areas based on provincial statistical units. During the four periods, the proportions of the three types of RF remained relatively stable in each province. Traditional agricultural provinces, such as Heilongjiang, Shandong, Henan, Sichuan and other provinces, had combined proportions of cores and edges exceeding 75%. These regions are the main grain production bases in China and have specific requirements for maintaining contiguous farmland. The RF types in the northwestern provinces, such as the Inner Mongolia Autonomous Region, Gansu and Xinjiang Uygur Autonomous Region, are mainly edge and scatter. Among the provinces in southwestern and southeastern China, such as Yunnan, Guangxi, Guangdong, Hunan, Fujian and Hainan, the proportion of scattered area is the highest, exceeding 50%. From 1980 to 2020, the Northeast and Northwest regions of China were the main areas of farmland restoration. The main method of farmland restoration in each province is expansion. The proportion of increment is relatively high in the northeast, northwest and southern regions. From 1990 to 2000, the GF area in 26 provinces exceeded 1000 square kilometers. This period was the most active time for farmland restoration. Shrinkage and subdivision are the main ways in which farmland is lost in various provinces. Perforation mainly occurs in provinces in Northeast China, North China and the Yangtze River Basin. From 1980 to 2020, the number of provinces with LF areas exceeding 10,000 square kilometers increased from 7 to 20. The scope of farmland loss in China has expanded over time.



(a) 1980–1990 (b) 1990–2000

(c) 2000–2010 (d) 2010–2020

Figure 5. *Cont.*

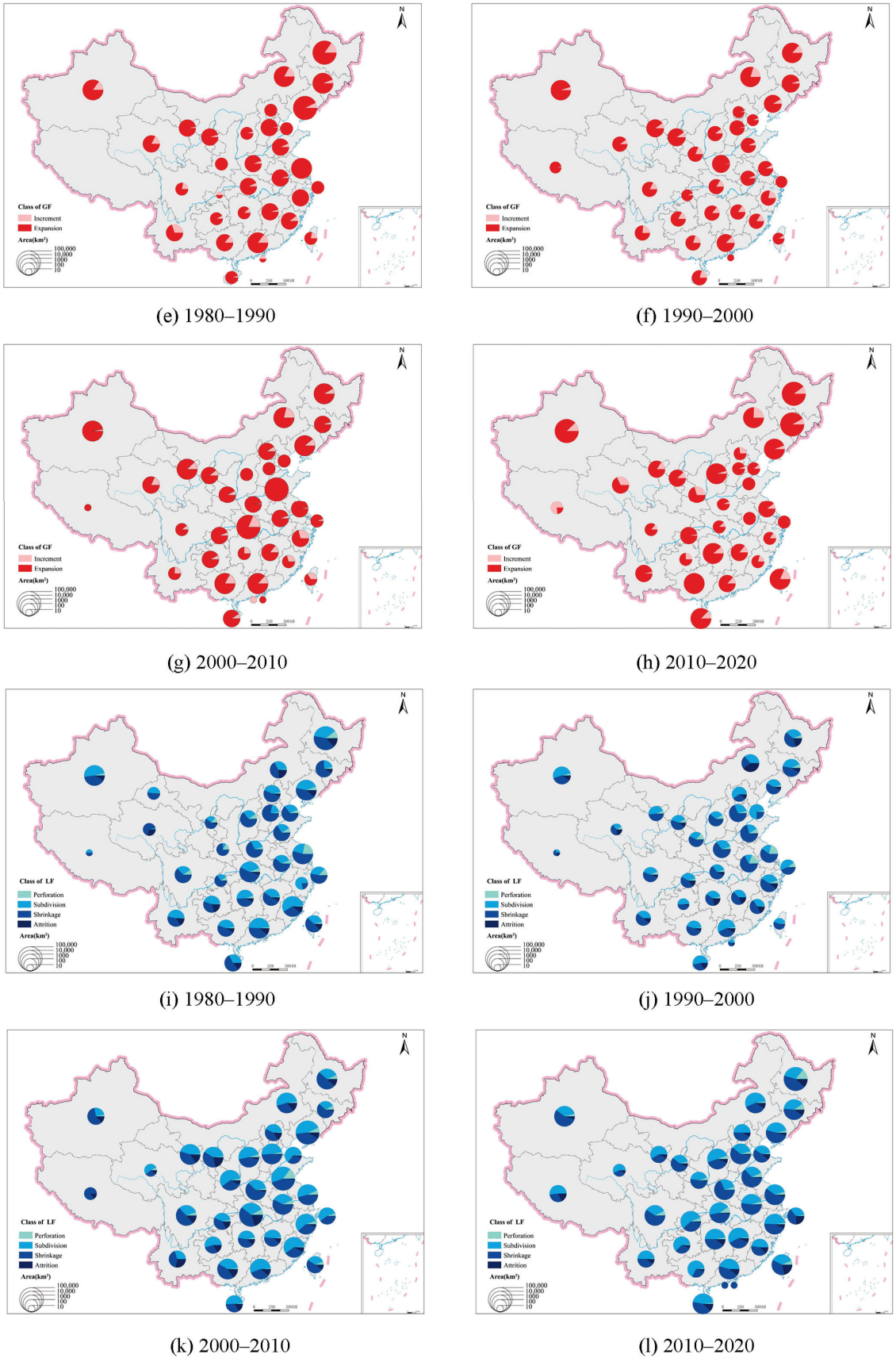
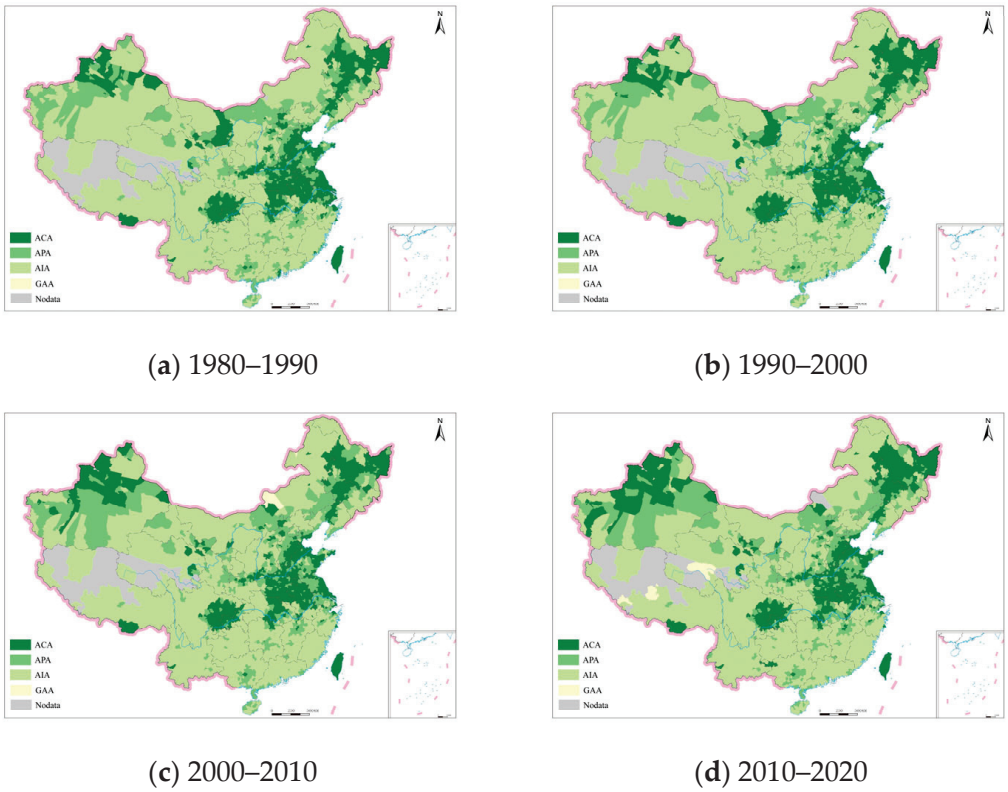


Figure 5. Statistics for the process of spatial change in farmland at the provincial level. (**a**–**d**) Pie charts of RF. (**e**–**h**) Pie charts of GF. (**i**–**l**) Pie charts of LF.

3.2. Zoning Based on the Dynamic Evolution Patterns of Farmlands

According to the combination of core, edge and scatter areas, China's agricultural counties were divided into agricultural core areas, agricultural protected areas, agricultural improvement areas and general agricultural areas. The results are shown in Figure 6. The Northeast Plain, North China Plain and Sichuan Basin are typical ACAs, which are related to the local topographical and geomorphological conditions. The APAs are mainly distributed around the core areas. Due to the large number of hills and mountains in Southern China, there is no scale advantage for agriculture in this area. With time, the ACAs and APAs expanded significantly, which was caused by local agricultural development policies.



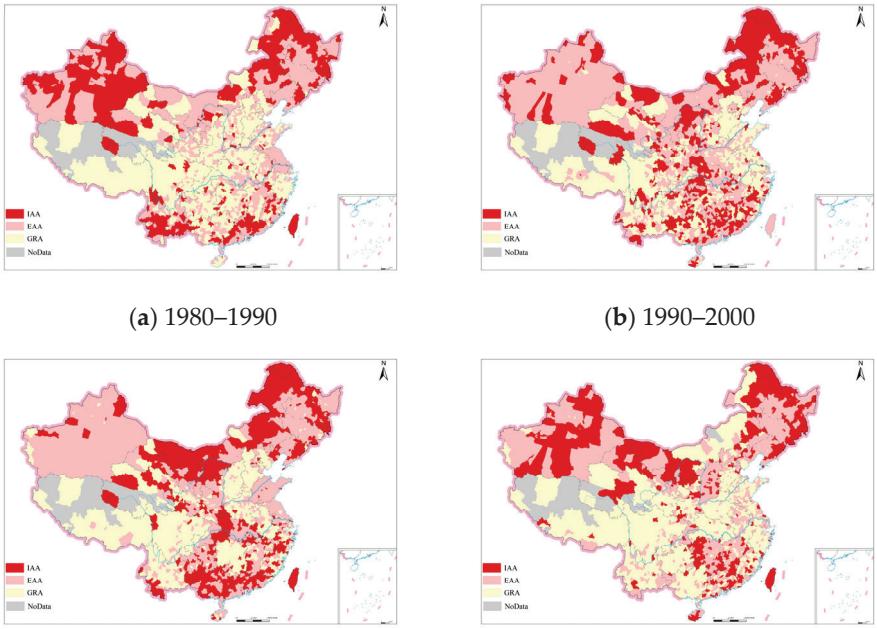
(**a**) 1980–1990 (**b**) 1990–2000

(**c**) 2000–2010 (**d**) 2010–2020

Figure 6. The results of agricultural scale zoning.

According to the difference between the proportion of increment and expansion, the agricultural counties in China were divided into increment advantage areas, expansion advantage areas and general restoration areas. The results are shown in Figure 7. The northwest, northeast and southeast regions of China were the regions with the greatest distributions of IAA and EAA. During the period from 1980 to 2020, the IAA and EAA regions in Central China first expanded and then contracted. The scales of IAAs and EAAs in the Yunnan–Guizhou Plateau gradually decreased.

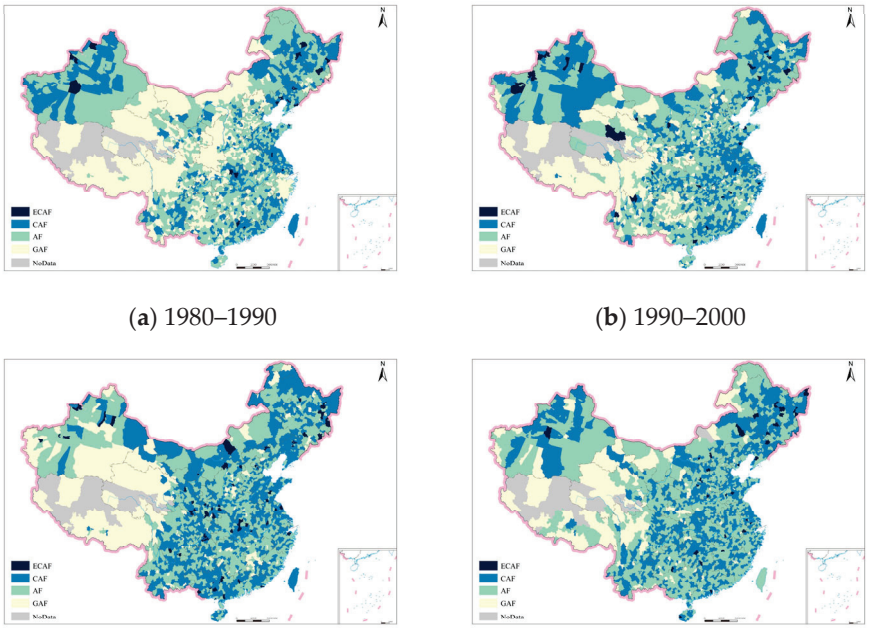
For the four loss processes based on area, the agricultural counties in China were divided into extremely complex areas of fragmentation, complex areas of fragmentation, fragmented areas and general areas of fragmentation. The fragmentation status of farmland in the four periods is shown in Figure 8. There were relatively few counties in China of the ECAF type; most were of the CAF and AF types. Over time, the scope of CAFs and AFs gradually expanded. By 2020, most counties in China showed farmland fragmentation. In traditional agricultural areas, such as the Northeast Plain, North China Plain and Sichuan Basin, there were many counties of the CAF type. This indicates that the fragmentation of farmland in traditional agricultural areas has complex characteristics.



(**a**) 1980–1990 (**b**) 1990–2000

(**c**) 2000–2010 (**d**) 2010–2020

Figure 7. The results of gaining farmland zoning.



(**a**) 1980–1990 (**b**) 1990–2000

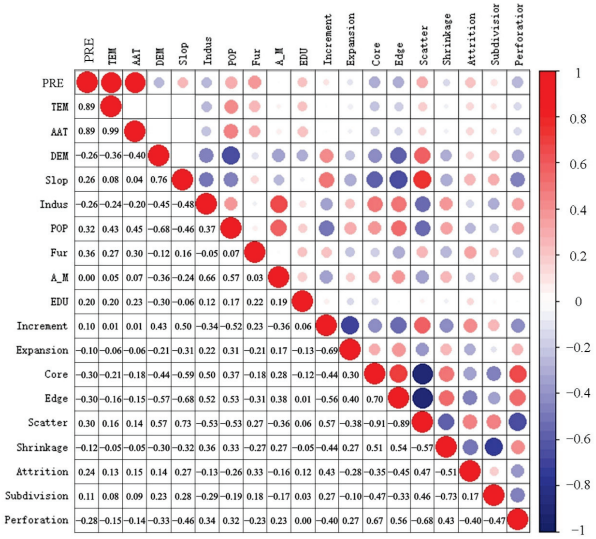
(**c**) 2000–2010 (**d**) 2010–2020

Figure 8. The results of fragmentation status.

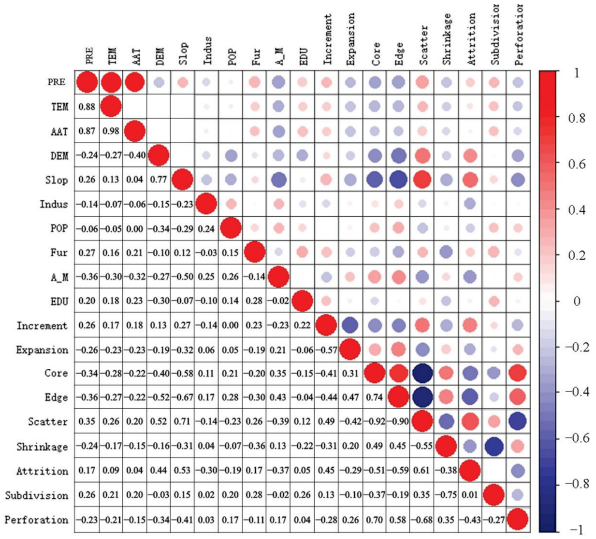
3.3. Correlation Analysis of Processes of Farmland Changes

Due to a lack of statistical data in earlier yearbooks, correlation analysis was carried out to analyze the process of spatial changes of farmland for 1990–2000, 2000–2010 and 2010–2020. The results are shown in Figure 9. During all time periods, the precipitation, temperature, accumulated temperature, fertilizer utilization efficiency and education level exhibited weak correlations with the various types of cultivated land. Terrain factors such as DEM and slope were positively correlated with increment, scatter, attrition and subdivision. The terrain factor was negatively correlated with expansion, core, edge, shrinkage and perforation. The levels of industrialization and population density were

negatively correlated with increment, scatter, attrition and subdivision but were positively correlated with expansion, core, edge, shrinkage and perforation. The level of agricultural mechanization was negatively correlated with increment, scatter and attrition but positively correlated with expansion, core and edge. The correlation coefficient values between farmland and related factors varied during different time periods. From 1990 to 2000, terrain factors had strong correlations with increment, core, edge, scatter and perforation. Industrialization and population density exhibited strong correlations with increment, core, edge and scatter. From 2000 to 2010, terrain factors showed strong correlations with core, edge, scatter and attrition. The correlations between industrialization, population density and farmland weakened. The correlations between agricultural mechanization level and core, edge and scatter grew stronger. From 2010 to 2020, terrain factors showed strong correlations with core, edge and scatter. There was almost no correlation between industrialization and farmland. The correlation between population density and edge and scatter grew stronger.

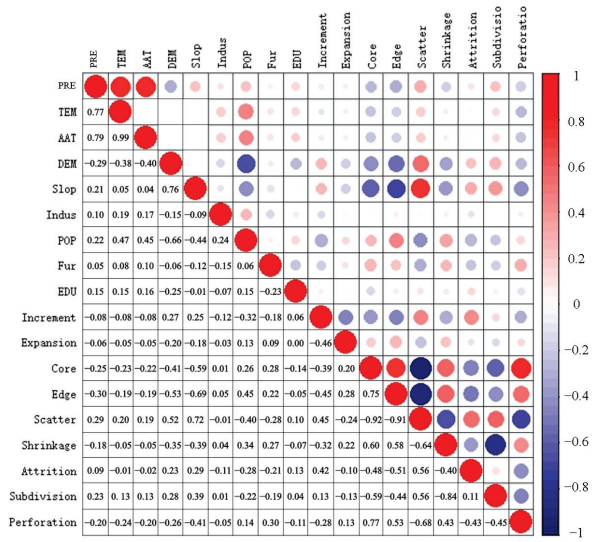


(**a**)



(**b**)

Figure 9. *Cont.*



(**c**)

Figure 9. Correlation analysis between variables. (**a**) 1990–2000; (**b**) 2000–2010; (**c**) 2010–2020.

4. Discussion

4.1. Spatiotemporal Distribution Pattern of Farmland Landscape Evolution

The remaining farmlands in China have a relatively stable distribution pattern, but there is still clear evolution in terms of farmland gains and losses. The remaining farmland is mainly edges and scattered areas. Although the core has the smallest proportion of area, there are central agricultural production areas, which is consistent with the current distribution of China's main agricultural production areas. Terrain factors significantly limit the distribution of remaining farmland. Core and edge are widely distributed in areas with low altitudes and small slopes, while scattered areas are more likely to be distributed in areas with slightly higher altitudes and steep slopes.

Expansion is the main means of farmland restoration. When a large number of expansions occurred in a previous period, these areas were often transformed into edge areas in the next period. The correlations between gained farmland and the factors selected in this article were weak, indicating that the process of farmland restoration is strongly related to governmental policies [35].

The fragmentation of farmland landscapes is a typical problem in agricultural development, especially in traditional Chinese agricultural production. With respect to farmland loss, the processes of subdivision and shrinkage accounted for more than 80% of the total lost land area and showed an increasing trend. This shows that the fragmentation of China's farmland increased during the study period. This finding is consistent with the results of other scholars [36]. The famous landscape ecologist Forman once proposed that the perforation process represents the beginning stage of landscape fragmentation. The process of subdivision and shrinkage is a manifestation of a further increase in the degree of fragmentation. Attrition is the final stage. From 1980 to 2020, the fragmentation process of farmland landscapes in China indicated mid-stage landscape evolution.

4.2. Policy Implications

Based on the patterns of farmland evolution, we should strictly adhere to the red line of 1.8 billion mu of cultivated land [37]. We need to ensure the scale advantage of the core and refer to edges to establish the protection boundary for core agricultural areas. We should firmly adhere to the strategy of balancing gains and losses of arable land. We can use measures such as supplementation and expansion to restore arable land, reduce the impact of shrinkage and subdivision on the scale of farmland and achieve a dynamic balance and sustainable development in agricultural spaces. There is much evidence that blindly expanding farmland causes great challenges for the protection of agricultural biodi-

versity [11]. Only when the agricultural ecosystem is healthy can agricultural production obtain the optimal effect. Therefore, farmland protection needs to pay equal attention to quantity, quality and ecology. Farmland with stable natural ecology that is suitable for crop growth should be chosen to supplement the amount of cultivated land.

According to the zoning results, there are significant differences in the evolution of farmland among different regions in China. The utilization of arable land in each region should be based on the local conditions. The scale advantage of core and edge areas in major agricultural production areas should be ensured, and farmland fragmentation due to land shrinkage, perforation and other factors should be prevented. We must adhere to the principle of "equal emphasis on quality" when adding advantageous areas of cultivated land. Pursuing only quantity goals while neglecting the quality of cultivated land may lead to increases in later stages.

From the correlation analysis results, it can be seen that the distribution of farmland should be determined based on the natural conditions and local socioeconomic factors. On the one hand, it is necessary to improve existing farmland. We need to level fields, improve the connectivity of cultivated land and enhance the suitability of agricultural production methods for cultivated land. On the other hand, we need to strictly implement the farmland supervision system and stop the conversion of farmland into nonagricultural land due to societal and economic development.

5. Conclusions

This study is based on a land cover dataset from 1980 to 2020, and landscape process modeling theory and related methods were used to reveal the spatiotemporal distribution patterns of farmland evolution in China. We used the advantage scale index to partition the evolution patterns of cultivated land. We analyzed the distribution patterns of cultivated land evolution based on natural and socioeconomic factors.

From 1980 to 2020, many types of farmland evolution patterns were found in China. The center of gravity of various cultivated lands has shifted significantly. The remaining farmland is mainly composed of edges and scatters, with core areas showing the lowest proportion. Farmland loss showed an increasing trend, with shrinkage and subdivision being the main types. Farmland evolution in China has long been in the mid-stage of fragmentation. The restoration of arable land has mainly occurred via expansion.

The agricultural core areas and agricultural protection areas were similar to the main agricultural production areas in China, and they were concentrated in the Northeast Plain, North China Plain, Sichuan Basin, the Middle and Lower Yangtze Valley Plain and other areas. The northwest and northeast regions have consistently been advantageous areas for increment and expansion. Complex areas of fragmented farmland are gradually expanding and are widely found in agricultural production areas, requiring key prevention measures.

The impact of different factors on the evolution of cultivated land varies. DEM and slope had a strong correlation with farmland evolution. In areas with low altitudes and small slopes, cores, edges, expansions and shrinkages were widely distributed. Increasing industrialization levels and a rapid increase in population led to an increase in the demand for food production in cities. Therefore, there was a greater distribution of farmland consisting of core and edge areas within regions.

The farmland process model used in this study can fully reflect the dynamic pattern of farmland changes over a period of time. The obtained results clearly demonstrate the spatial differences in the regional clustering of farmland evolution processes. A discussion of the relevant factors involved in the different processes of change is conducive to understanding the internal mechanism of farmland change and has practical significance for maintaining a dynamic balance on farmland. Because of the difficulties associated with data acquisition, some limitations exist in this study, especially in terms of the discussion of relevant factors involved in farmland changes. The factors affecting changes in farmland are complex. The current summary is not comprehensive, and it is necessary to quantify more factors for better correlation analysis.

Author Contributions: Conceptualization, Lili Jiang and Yan Fu; methodology, Yan Fu and Lili Jiang; software, Yan Fu and Yapeng Zhao; validation, Yan Fu, Yapeng Zhao and Lili Jiang; formal analysis, Yan Fu and Qingwen Qi; investigation, Yan Fu and Qingwen Qi; resources, Qingwen Qi and Lili Jiang; data curation, Yan Fu and Yapeng Zhao; writing—original draft preparation, Yan Fu and Yapeng Zhao; writing—review and editing, Lili Jiang and Yan Fu; visualization, Yan Fu and Yapeng Zhao; supervision, Lili Jiang and Qingwen Qi; project administration, Lili Jiang and Qingwen Qi; funding acquisition, Lili Jiang. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Third Xinjiang Scientific Expedition Program, grant number 2021xjkk0303.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Acknowledgments: The authors thank the reviewers and editors for their valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cheng, L.; Jiang, P.; Chen, W.; Li, M.; Wang, L.; Gong, Y.; Pian, Y.; Xia, N.; Duan, Y.; Huang, Q. Farmland protection policies and rapid urbanization in China: A case study for Changzhou City. *Land Use Policy* **2015**, *48*, 552–566. [CrossRef]
2. Lichtenberg, E.; Ding, C. Assessing farmland protection policy in China. *Land Use Policy* **2008**, *25*, 59–68. [CrossRef]
3. Xin, L.; Li, X. China should not massively reclaim new farmland. *Land Use Policy* **2018**, *72*, 12–15. [CrossRef]
4. Fu, M.; Hu, Z.; Wu, G. Analysis of evolutionary law rule of farmland landscape. *Trans. Chin. Soc. Agric. Eng.* **2005**, *21*, 54–58.
5. Jiang, P.; Li, M.; Cheng, L. Dynamic response of agricultural productivity to landscape structure changes and its policy implications of Chinese farmland conservation. *Resour. Conserv. Recycl.* **2020**, *156*, 104724. [CrossRef]
6. Chen, W.X.; Wang, G.Z.; Cai, W.N.; Che, X.Y.; Zhou, W.H.; Zhang, C.; Zeng, J. Spatiotemporal mismatch of global grain production and farmland and its influencing factors. *Resour. Conserv. Recycl.* **2023**, *194*, 107008. [CrossRef]
7. Li, T.T.; Long, H.L.; Zhang, Y.N.; Tu, S.S.; Ge, D.Z.; Li, Y.R.; Hu, B.Q. Analysis of the spatial mismatch of grain production and farmland resources in China based on the potential crop rotation system. *Land Use Policy* **2017**, *60*, 26–36. [CrossRef]
8. Ma, Q.; Han, W.T.; Huang, S.J.; Dong, S.D.; Li, G.; Chen, H.P. Distinguishing Planting Structures of Different Complexity from UAV Multispectral Images. *Sensors* **2021**, *21*, 1994. [CrossRef] [PubMed]
9. Ng, K.; Barton, P.S.; Blanchard, W.; Evans, M.J.; Lindenmayer, D.B.; Macfadyen, S.; McIntyre, S.; Driscoll, D.A. Disentangling the effects of farmland use, habitat edges, and vegetation structure on ground beetle morphological traits. *Oecologia* **2018**, *188*, 645–657. [CrossRef] [PubMed]
10. Yu, P.H.; Fennell, S.; Chen, Y.Y.; Liu, H.; Xu, L.; Pan, J.W.; Bai, S.Y.; Gu, S.X. Positive impacts of farmland fragmentation on agricultural production efficiency in Qilu Lake watershed: Implications for appropriate scale management. *Land Use Policy* **2022**, *117*, 106108. [CrossRef]
11. Clough, Y.; Kirchweger, S.; Kantelhardt, J. Field sizes and the future of farmland biodiversity in European landscapes. *Conserv. Lett.* **2020**, *13*, e12752. [CrossRef]
12. Siljander, M.; Kuronen, T.; Johansson, T.; Munyao, M.N.; Pellikka, P.K.E. Primates on the farm—Spatial patterns of human-wildlife conflict in forest-agricultural landscape mosaic in Taita Hills, Kenya. *Appl. Geogr.* **2020**, *117*, 102185. [CrossRef]
13. Liu, X.N.; Zhang, W.W.; Li, H.; Sun, D.F. Modeling patch characteristics of farmland loss for site assessment in urban fringe of Beijing, China. *Chin. Geogr. Sci.* **2013**, *23*, 365–377. [CrossRef]
14. Zou, L.L.; Liang, Y.F.; Hu, X.D.; Bai, M.D.; Wen, Q. A multidimensional assessment and analysis of farmland fragmentation in the southeast coastal area of China. *Land Degrad. Dev.* **2023**, *34*, 4021–4034. [CrossRef]
15. Finch, T. Restoring farmlands for food and nature. *One Earth* **2020**, *3*, 665–668. [CrossRef]
16. Yu, Y.; Zhang, J.; Zhang, K.; Xu, D.; Qi, Y.; Deng, X. The impacts of farmer ageing on farmland ecological restoration technology adoption: Empirical evidence from rural China. *J. Clean. Prod.* **2023**, *430*, 139648. [CrossRef]
17. Cheng, L.; Xia, N.; Jiang, P.; Zhong, L.; Pian, Y.; Duan, Y.; Huang, Q.; Li, M. Analysis of farmland fragmentation in China Modernization Demonstration Zone since "Reform and Openness": A case study of South Jiangsu Province. *Sci. Rep.* **2015**, *5*, 11797. [CrossRef] [PubMed]
18. Fu, M.; Hu, Z.; Wu, G. Analysis of farmland landscape pattern evolvement rule on Zhongjie Friendship Farm. *Trans. Chin. Soc. Agric. Eng.* **2006**, *22*, 63–67.
19. Jiao, F.; Wen, Z.; Wang, F.; Zhang, X.; Yang, Q.; Li, R. Analysis of Landscape Structure at County Scale in Loess Hilly Region. *J. Soil Water Conserv.* **2005**, *19*, 167–170.
20. Liu, J.; Jin, X.; Xu, W.; Sun, R.; Han, B.; Yang, X.; Gu, Z.; Xu, C.; Sui, X.; Zhou, Y. Influential factors and classification of cultivated land fragmentation, and implications for future land consolidation: A case study of Jiangsu Province in eastern China. *Land Use Policy* **2019**, *88*, 104185. [CrossRef]
21. Liu, X.; Bian, Z.; Sun, Z.; Wang, C.; Sun, Z.; Wang, S.; Wang, G. Integrating Landscape Pattern Metrics to Map Spatial Distribution of Farmland Soil Organic Carbon on Lower Liaohe Plain of Northeast China. *Land* **2023**, *12*, 1344. [CrossRef]

22. Xu, W.; Jin, X.; Liu, J.; Zhou, Y. Impact of cultivated land fragmentation on spatial heterogeneity of agricultural agglomeration in China. *J. Geogr. Sci.* **2020**, *30*, 1571–1589. [CrossRef]
23. Zhang, J.; Rao, W.; Xiao, H.; Zhou, Q. Change of farmland landscape pattern in urbanization process based on RS and GIS technologies. *J. South China Agric. Univ. Nat. Sci. Ed.* **2005**, *26*, 106–109.
24. Zhang, Y.; Sharma, S.; Bista, M.; Li, M. Characterizing changes in land cover and forest fragmentation from multitemporal Landsat observations (1993–2018) in the Dhorpatan Hunting Reserve, Nepal. *J. For. Res.* **2022**, *33*, 159–170. [CrossRef]
25. Li, S.; Yang, B. Introducing a new method for assessing spatially explicit processes of landscape fragmentation. *Ecol. Indic.* **2015**, *56*, 116–124. [CrossRef]
26. Li, X.; Weng, W.; Li, M. Assessing natural forest fragmentation process dynamics and its drivers in the Pacific northwest region, USA. *J. Nanjing For. Univ. Nat. Sci. Ed.* **2021**, *45*, 174–182.
27. Lu, Y.; Wang, Z.; Xia, X.; Yuan, H.; Li, M.; Xu, W. Spatiotemporal evolutions and spatial processes of cultivated land landscape in Daxinganling forest-grass ecotone in the Inner Mongonia. *Acta Ecol. Sin.* **2023**, *43*, 1209–1218.
28. Zhang, Y.; Li, X.; Li, M. Assessing Spatio-Temporal Variation and Associated Factors of Forest Fragmentation from Morphological Spatial Pattern Analysis and Geo-Detector Analyses: A Case Study of Xinyu City, Jiangxi Province of Eastern China. *Forests* **2023**, *14*, 2376. [CrossRef]
29. Baek, S.; Yoon, H.; Hahm, Y. Assessment of spatial interactions in farmland abandonment: A case study of Gwangyang City, Jeollanam-do Province, South Korea. *Habitat Int.* **2022**, *129*, 102670. [CrossRef]
30. Chang, H.S.; Chen, T.L. Exploring spatial patterns of farmland transactions and farmland use changes. *Environ. Monit. Assess.* **2015**, *187*, 596. [CrossRef]
31. Li, D.J.; Zhang, H.Q.; Xu, E.Q. A spatial directivity-based sensitivity analysis to farmland quality evaluation in arid areas. *Environ. Sci. Pollut. Res.* **2022**, *29*, 66359–66372. [CrossRef] [PubMed]
32. Thompson, A.W.; Prokopy, L.S. Tracking urban sprawl: Using spatial data to inform farmland preservation policy. *Land Use Policy* **2009**, *26*, 194–202. [CrossRef]
33. Jiang, Q.; Zhao, Z.; Mu, Y.; Zhao, D.; Wu, B. A study on the agglomeration characteristics of the scallop aquaculture industry in China. *Isr. J. Aquac. Bamidgeh* **2022**, *74*, 1–14. [CrossRef]
34. Zhang, W.-Y.; Wei, Z.-W.; Wang, B.-H.; Han, X.-P. Measuring mixing patterns in complex networks by Spearman rank correlation coefficient. *Phys. A Stat. Mech. Appl.* **2016**, *451*, 440–450. [CrossRef]
35. Cao, Y.; Kong, L.; Zhang, L.; Ouyang, Z. Spatial characteristics of ecological degradation and restoration in China from 2000 to 2015 using remote sensing. *Restor. Ecol.* **2020**, *28*, 1419–1430. [CrossRef]
36. Gu, T.; Chen, W.; Liang, J.; Pan, S.; Ye, X. Identifying the driving forces of cultivated land fragmentation in China. *Environ. Sci. Pollut. Res.* **2023**, *30*, 105275–105292. [CrossRef] [PubMed]
37. Men, K.; Zhu, H.; Jiang, L. Can the Baseline of 1.8 Billion Mu Cultivated Land be Retained in China?—A Macro Approach to Land Reservation from Improved-MGM Model. In Proceedings of the Conference of the International-Institute-of-Applied-Statistics-Studies, Yantai, China, 13–15 August 2008; pp. 1565–1569.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

remote sensing

MDPI

Article

Classification of Lakebed Geologic Substrate in Autonomously Collected Benthic Imagery Using Machine Learning

Joseph K. Geisz [1], Phillipe A. Wernette [1,*] and Peter C. Esselman [2]

[1] Michigan Technological University, Great Lakes Research Center, Contractor to the US Geological Survey, Houghton, MI 49931, USA; jgeisz@contractor.usgs.gov
[2] US Geological Survey, Great Lakes Science Center, Ann Arbor, MI 48105, USA; pesselman@usgs.gov
* Correspondence: pwernett@mtu.edu

Abstract: Mapping benthic habitats with bathymetric, acoustic, and spectral data requires georeferenced ground-truth information about habitat types and characteristics. New technologies like autonomous underwater vehicles (AUVs) collect tens of thousands of images per mission making image-based ground truthing particularly attractive. Two types of machine learning (ML) models, random forest (RF) and deep neural network (DNN), were tested to determine whether ML models could serve as an accurate substitute for manual classification of AUV images for substrate type interpretation. RF models were trained to predict substrate class as a function of texture, edge, and intensity metrics (i.e., features) calculated for each image. Models were tested using a manually classified image dataset with 9-, 6-, and 2-class schemes based on the Coastal and Marine Ecological Classification Standard (CMECS). Results suggest that both RF and DNN models achieve comparable accuracies, with the 9-class models being least accurate (~73–78%) and the 2-class models being the most accurate (~95–96%). However, the DNN models were more efficient to train and apply because they did not require feature estimation before training or classification. Integrating ML models into benthic habitat mapping process can improve our ability to efficiently and accurately ground-truth large areas of benthic habitat using AUV or similar images.

Keywords: remote sensing; machine learning; benthic habitat mapping; autonomous underwater vehicle; underwater photography

1. Introduction

A major goal of underwater benthic mapping is to describe and map geologic substrates in a manner that improves our understanding of the relation between the physical properties of sea-, lake-, or riverbed environments and the species that use them. Here, a geologic substrate is defined as "... a surface or volume of sediment or rock where physical, chemical, and biological processes occur" [1]. Swath sonar and aerial remote sensing methods have made it possible to map bathymetric, acoustic, and spectral proxies of geologic substrates with continuous coverage at high resolutions and over large spatial extents [2–5]. However, conversion of bathymetric, acoustic, or spectral proxy data into maps of geologic substrate requires interpretation and analysis relative to georeferenced ground truth observations. Regardless of the specific analytical approach used to produce substrate maps, the quality of the maps will always benefit from robust ground truth data.

Ground truth observations can consist of physical samples of sediments, photographs, or video footage of benthic environments [6]. Requirements for spatial accuracy, replication, and post-processing of ground truth observations depend on mapping objectives and context. Visual imaging of the benthic environment using still or video cameras has become a particularly important source of ground truth information, with clear applications for the interpretation of geologic substrates [4]. A distinct advantage of visual imagery as a ground truth source for interpreting bathymetric, acoustic, or spectral data is that still and/or video

images can be gathered quickly over relatively large areas, particularly when cameras are deployed from autonomous underwater vehicles (AUVs) that can travel tens or hundreds of kilometers on a single battery charge gathering images continuously [3,7]. However, acquisition of imagery in a way that encompasses characteristic variation in substrates is only part of the challenge. Effective use of imagery in substrate mapping may also require post-processing of imagery (i.e., corrections for over/underexposure, color, lighting, etc.) to make them more visually interpretable, assignment of substrate classes to imagery, and supervised classification to produce predictive maps across continuous areas. The assignment of substrate classes to imagery at specific locations (i.e., image "annotation") can be particularly time consuming if it requires human observer(s) to evaluate each image or video clip to determine the substrate class(es) present [8,9]. Automation of class assignments using computational methods has potential to both accelerate the preparation of ground truth data and to increase the number of ground truth observations available for supervised classification [5,9,10]. Here, we focus on the challenge of automated image annotation for geologic substrate classes in still images gathered from AUVs.

An important consideration for classifying any data is the scheme to be used (i.e., the definitions of the substrate classes to be assigned to the ground truth data and eventually mapped to large areas). Bathymetry classification schemes vary by application, image type, and research objective. Numerous schemes have been proposed and utilized for this purpose, including the Wentworth grade scale [11], Trefethen's classification scheme [12], Shepard's classification scheme [13,14], and Folk's classification scheme [15]. National Oceanographic and Atmospheric Administration's (NOAA) Coastal and Marine Ecological Classification Standard (CMECS) utilizes the Wentworth grade scale for geologic substrates in a broader classification framework, describing coastal and marine environments based on four components: water column, biotic, substrate, and geoform [16]. Each of these components can have multiple modifiers and come together to form a biotope (i.e., combination of abiotic features and associated species). Mapping the geologic substrate from imagery specifically focuses on the CMECS geologic substrate component at fine spatial scales (100 to 101 sq m image footprint). CMECS provides a well-established framework to classify benthic substrate in marine environments that is the standard for many U.S. federal agencies [5,17].

Annotation of substrates in benthic images can be done manually, but the process of interpreting each image is time-intensive and thus not scalable for larger datasets [8]. Depending on the scale of the project and amount of imagery gathered, image annotation can consist of assigning one or more substrate classes to an entire image, identifying objects within the image, or segmenting the image into multiple classes on a pixel-by-pixel basis. While manual annotation of classes in whole images may be more straightforward than automated object detection or image segmentation, it is only likely to be suitable for datasets up to several hundred or thousands of images. For datasets containing hundreds of thousands or more images or video frames, such as those produced by AUVs, the effort involved in human-labeling becomes unwieldy or impossible. We propose that a more efficient alternative to manual substrate classification is through the development and application of machine learning (ML) models [18–24]. Machine learning provides the opportunity to efficiently classify points, parts of an image, or whole images based on underlying mathematical relationships within a set of training data. When developed and trained appropriately, ML models can help efficiently and accurately classify whole images and parts of images in large datasets that were previously intractable with manual labeling [25]. ML classification of geologic substrates, therefore, offers the possibility to greatly densify ground truth observations when used in conjunction with image acquisition technologies with large spatial ranges and frequent sampling.

The use of ML to classify benthic imagery is a well-explored topic [5,10,25–28]. Classifying an entire image into pre-defined categories such as CMECS substrate classes requires some degree of a supervised learning approach. Supervised learning algorithms are trained using datasets that have input and output pairs. For instance, an input could be an AUV

image, and an output is that image's substrate class. The accuracy of the resulting algorithm depends both on the quality of the labeled dataset and the model architecture.

Machine learning is a broad term that includes a wide range of algorithms that use training data to learn patterns and make predictions from those patterns. There are tradeoffs in choosing which model to use. Algorithms vary in terms of the number of hand-tuned parameters required, the ability to discern abstract patterns from low-level data, and the computational complexity both during training and application. Conventional ML models, such as support vector machines (SVMs), often are easier to interpret and understand and have been used to classify benthic habitat in the Laurentian Great Lakes [5] and elsewhere [25,29]. Reif and others used SVMs to classify a relatively small area (11.7 sq. km.) of southwestern Lake Michigan into "Tier I" and "Tier II" biogeological classes, where the two tiers are hierarchically nested and based on CMECS biogeological classifications, with an accuracy greater than 89%, demonstrating the direct application of ML techniques to more efficiently and accurately map biogeological classes [5]. However, as more and more data are collected, it becomes increasingly important to explore alternative ML models that are more scalable and require less subjective tuning.

The purpose of this paper is to explore the utility of two different ML approaches for assigning geologic substrate classes to whole AUV images for benthic habitat mapping—random forest (RF) and deep neural network (DNN) models. Random forest and DNN models were developed to classify images into a 9-class scheme adapted from the CMECS substrate standard for use with visual image interpretation in lieu of physical grab samples. Under this approach, each image receives a single substrate class assignment. The 9-class training data were also aggregated to 6-class and 2-class schemes to explore the impact of the number of classes on model accuracy. Model accuracy and the computational cost of model training and imputation were compared between the two ML models on a desktop PC with 2 Intel Xeon Gold 6238R CPUs with 28 cores each, 64 GB of RAM, and an NVIDIA RTX A6000 GPU. Comparison of the required data pre-processing steps for each model type is also discussed, as it provides valuable context about required resources and can be used to guide future model development.

2. Materials and Methods

2.1. Description of Dataset

All supervised ML models must be trained using a dataset where every input, in this case each image, has already been classified. The training processes search for the model parameters that minimize error in the prediction of these training data. Often, this process is stochastic and iterative using techniques like stochastic gradient descent or Adam [30] to optimize some loss function. Model accuracy is determined by making a prediction from the trained model to another labeled dataset (i.e., validation data) to test how effective the trained parameters generalize to unseen data.

Creating the training and validation datasets required a manual labeling effort and some important choices. Ideally, the model would be trained on images encompassing the full range of optical conditions encountered by the AUV across the study area, inclusive of variation driven by ambient light levels, water hues, and the appearance of the substrate itself. For instance, an image of fine sediment on a sunny day in clear shallow water should be classified the same as an image of fine sediment on a dark, overcast day in deep, dark, turbid water. Having high levels of variation in the training data should allow the models to discern which features are important in differentiating between classes. This also helps to avoid the problem of overfitting—where a model effectively "memorizes" a dataset and thus predicts extremely well on the training data but very poorly on unseen data. The overarching goal of model development was to produce trained models that were generalizable.

2.1.1. AUV Image Dataset

An L3Harris-Ocean Server Iver3 autonomous underwater vehicle (AUV) was used to collect nadir images of the lakebed at a rate of five images per second at a forward velocity of 1.5 knots (0.77 m/s). The Iver3 was modified to carry a custom camera payload that included an 8.95-megapixel color machine vision camera (Allied Vision Manta 895C with a 10 mm lens) and a Nerian brand Karmin2, 10 cm baseline, grayscale stereo camera (carrying two 2.0-megapixel E2v EV76C570 grayscale sensors with 10 mm lenses) triggered by a Nerian SceneScanPro FPGA stereo vision sensor (sourced from Stadtroda, Germany). The SceneScanPro produced real-time stereo disparity maps and elevation point clouds at 4 frames per second. An iXBlue Phins Compact C3 inertial navigation system (INS) onboard the Iver3 provided real-time estimates of latitude and longitude, with a horizontal error less than 0.3% of distance traveled (i.e., 3 m error per 1000 m traveled).

Approximately 2.6 million geotagged still images were acquired across more than 400 km of transect distance traveled by the AUV in the nearshore environments of all five Laurentian Great Lakes during the timeframe of this study. These images represent a valuable source of georeferenced substrate ground truth data. The greatest challenge with using these data for ground truth is annotating images with known or predicted substrate classes. Annotating millions of images manually for substrate type would not only be prohibitively time intensive, but would also require a large group of experienced image annotation specialists with a high degree of label agreement. Automated classification using a machine learning (ML) model offers the potential to fully utilize this dataset by automating the image annotation task.

Substrate prediction models were primarily trained from the RGB intensity data from the color camera or their derivatives, but other sensor data on board the AUV can provide useful information. The point clouds from the stereo camera contain information about the roughness of the lakebed. The first and second principal components of the point cloud, excluding outliers, define the plane of best fit for the points. The standard deviation of the points' heights off this plane (i.e., along the third principal component, orthogonal to the others) was calculated and provided a proxy for the roughness of the surface. This was used as additional information for the RF classifiers. Henceforth, this calculated point cloud standard deviation will be referred to as the "plane standard deviation." Plane standard deviation was calculated as a time series throughout each AUV mission, and a plane standard deviation value was interpolated for color images, serving as an approximation of surface roughness below the AUV when a given color image was taken. The interpolation step was necessary because the color and stereo cameras were not synchronized but imaging independent of one another.

2.1.2. Classification Scheme

The Coastal and Marine Ecological Classification Standard (CMECS) defines substrate as "the non-living materials that form an aquatic bottom or seafloor, or that provide a surface (e.g., floating objects, buoys) for growth of attached biota." [16]. The Substrate Component of CMECS contains three "origins"—geologic, biogenic, and anthropogenic. Anthropogenic substrates are rare in our images, although biogenic origin substrates, such as shell hash, were present. Here, the focus was on substrates of geologic origin only. Within geologic substrates there are 2 classes: rock substrate and unconsolidated mineral substrate. Rock substrate is divided into bedrock and megaclast substrates. Unconsolidated mineral substrate is divided into coarse and fine unconsolidated substrate subclasses, which are further characterized by groups and subgroups, with empirical breakpoints between classes based on the maximum dimension of substrate particles (Table 1).

Table 1. CMECS hierarchy for substrates within the Geologic Origin CMECS class descriptions [16]. Size cutoffs for the substrate classes discussed in the paper are presented in parentheses.

CMECS Substrate Class	CMECS Substrate Subclass	CMECS Substrate Group	CMECS Substrate Subgroup	Label
Consolidated Mineral	Bedrock			Bedrock
	Megaclast			(>4096 mm)
Unconsolidated Mineral	Coarse Unconsolidated	Gravel	Boulder	Boulder (256 mm to <4096 mm)
			Cobble	Cobble (64 mm to <256 mm)
			Pebble	Pebble (4 mm to <64 mm)
			Granule	Granule (2–4 mm)
		Gravel Mixes	Sandy Gravel	Gravel Mixes
			Muddy Sandy Gravel	
			Muddy Gravel	
		Gravelly	Gravelly Sand	Gravelly
			Gravelly Muddy Sand	
			Gravelly Mud	
	Fine Unconsolidated	Slightly Gravelly	Slightly Gravelly Sand	Slightly Gravelly
			Slightly Gravelly Muddy Sand	
			Slightly Gravelly Sandy Mud	
			Slightly Gravelly Mud	
		Sand	Very Coarse Sand	Fine (<2 mm)
			Coarse Sand	
			Medium Sand	
			Fine Sand	
			Very Fine Sand	
		Muddy Sand	Silty Sand	
			Silty-Clayey Sand	
			Clayey Sand	
		Sandy Mud	Sandy Silt	
			Sandy Silt-Clay	
			Sandy Clay	
		Mud	Silt	
			Silt-Clay	
			Clay	

CMECS uses a modified version of Wentworth [11] mineral grain size descriptors for categorizing particles by diameter (Table 1). For example, gravel particles have a maximum dimension of between 2 mm and 4096 mm, and particles smaller than this are considered sand, mud, clay, or silt. A substrate within the Gravel group contains greater than 80% gravel particles, with subgroups differentiated by the "median Gravel size". However, this classification was problematic for classifying sediment size and substrate classification

in AUV images because any substrates not directly visible at the lakebed surface were impossible to quantify.

Not collecting sediment grabs limits the ability for characterizing the volumetric distribution and median sizes of different sediment classes, but collecting AUV images is more efficient and less invasive, meaning larger areas can be mapped more quickly. Calculating the median particle size from a typical image would require measuring every individual particle present. However, variations in environmental (water clarity/turbidity), camera hardware (lens focal length, exposure settings, sensor resolution), and AUV position (altitude above lakebed, roll, and pitch) combined to limit our ability to resolve particles finer than 2 mm. These limitations affected the classification process in two important ways. First, all substrates finer than 2 mm were attributed to a single "Fine" class that combines sand, mud, and their mixes. Second, classes were determined based on areal coverage of particle size groups in an image, rather than by calculating the median particle size. On the upper end of the particle size gradient, it was not possible to distinguish subclasses in the rock substrate class because megaclast particles are greater than 4.0 m in diameter, which is larger than the 1.95 m^2 imaged area from the AUV at 1.75 m altitude. Due to the limitations inherent to image-based classification, the CMECS substrate scheme was simplified to only those classes that could be reliably resolved, leading to sum aggregation at the subgroup level (see "label" column of Table 1), and the definitions for each class were defined relative to the aerial coverage of different particle sizes in whole images (Table 2).

Table 2. Substrate classes, abbreviations used for each throughout this paper (in parentheses), and their definitions. Dominant particle by areal extent is used rather than median particle size as defined in CMECS.

Label (Abbreviation)	Image Class Definition
Bedrock (Be)	The substrate in the image belongs to the Rock CMECS class, either bedrock or megaclast. This is a substrate with continuous formations of bedrock or megaclast (particles $\geq$ 4.0 m) that cover 50% or more of the image surface.
Boulder (Bo)	The substrate in the image belongs to the CMECS Boulder Subgroup. The Geologic Substrate contains >80% Gravel, with the areal extent dominated by Gravel particles of size 256 mm to <4096 mm.
Cobble (Co)	The substrate in the image belongs to the CMECS Cobble Subgroup. The Geologic Substrate contains >80% Gravel, with the areal extent dominated by Gravel particles of size 64 mm to <256 mm
Pebble (Pe)	The substrate in the image belongs to the CMECS Boulder Subgroup. The Geologic Substrate contains >80% Gravel, with the areal extent dominated by Gravel particles of size 4 mm to <64 mm.
Granule (Gran)	The substrate in the image belongs to the CMECS Boulder Subgroup. The Geologic Substrate contains >80% Gravel, with the areal extent dominated by Gravel particles of size 2 mm to <4 mm.
Gravel Mixes (GM)	The substrate in the image belongs to the CMECS Gravel Mixes Group. The Geologic Substrate surface layer contains 30% to <80% Gravel (particles 2 mm to <4096 mm).
Gravelly (Gr)	The substrate in the image belongs to the CMECS Gravelly Group. The Geologic Substrate surface layer contains 5% to <30% Gravel (particles 2 mm to <4096 mm).
Slightly Gravelly (SGr)	The substrate in the image belongs to the CMECS Slightly Gravelly Group. The Geologic Substrate surface layer contains from a trace (0.01%) of Gravel to 5% Gravel (particles 2 mm to <4096 mm).
Fine (F)	The substrate in the image belongs to the CMECS Fine Unconsolidated Substrate Subclass, but not the Slightly Gravelly Group. The Geologic Substrate surface layer contains no trace of Gravel and is composed entirely of particles <2 mm, including sand, mud (clay and silt), and mixed types.

A variety of factors made it challenging to classify some images. For instance, benthic algae sometimes partially or completely obscured the underlying substrate in many of the images. When less than 50% of the image was obscured by algae, the remaining visible substrate was used to classify the image. However, it was common for images to be more than 50% covered by algae. If the obscured substrate, inferred underneath the algae, could not be reliably associated with a coarse unconsolidated substrate subclass, then the image was classified as 'Coarse Algae' (i.e., coarse unconsolidated substrates were present, but not able to be attributed with a subclass label). In some cases, it was completely impossible to determine the image substrate class, in which case the image was classified as 'Unknown'. This was often caused by poor lighting conditions (i.e., underexposure or overexposure), water turbidity, image blur, co-dominant classes, or another inability to decipher the image. Images labeled as Unknown were not used in any of the ML models.

All images in the AUV training dataset were classified according to the 9 classes listed in the Label column of Table 1, in addition to Coarse Algae and Unknown. However, only the 9 classes listed in Table 1 were used in the substrate classification model development. The 9-class scheme was further aggregated to 6-and 2-class schemes (Table 3). The 6-class scheme was based on the perceived functional role of similar substrate classes to species. For instance, boulder and cobble may serve similar functions for spawning fishes in the Laurentian Great Lakes and thus can be combined into a single "very coarse" class. Bedrock is considered consolidated, boulder and cobble are combined into a very coarse class, pebble and granule become moderately coarse, and gravelly and gravel mix are mixed coarse (Table 3). The 2-class scheme further simplified class labels by aggregating boulder, cobble, pebble, granule, gravel mixes, and gravelly to a single "coarse" class, while slightly gravelly and fine were mapped to a single "fine" class. In addition, the coarse algae images were included in the coarse class in the 2-class scheme.

Table 3. Image annotation scheme modified from CMECS, with the number of images in each class in parentheses. Images were annotated at the 9-class level and subsequently aggregated into 6-class and 2-class schemes.

<table>
<tr><th>9-Class</th><th>6-Class</th><th>2-Class</th></tr>
<tr><td>Bedrock (Be)
(21)</td><td>Consolidated (Con)
(21)</td><td rowspan="8">Coarse * (C)
(3497)</td></tr>
<tr><td>Boulder (Bo)
(894)</td><td rowspan="2">Very Coarse (VC)
(1632)</td></tr>
<tr><td>Cobble (Co)
(738)</td></tr>
<tr><td>Pebble (Pe)
(83)</td><td rowspan="2">Moderately Coarse (MoC)
(90)</td></tr>
<tr><td>Granule (Gran)
(7)</td></tr>
<tr><td>Gravel Mix (GM)
(293)</td><td rowspan="2">Mixed Coarse (MiC)
(370)</td></tr>
<tr><td>Gravelly (Gr)
(77)</td></tr>
<tr><td>Slightly Gravelly (SGr)
(96)</td><td>Mixed (M)
(96)</td><td rowspan="2">Fine (F)
(1438)</td></tr>
<tr><td>Fine (F)
(1342)</td><td>Fine (F)
(1342)</td></tr>
<tr><td colspan="3">Coarse Algae (CA) (1405)</td></tr>
</table>

* The 2-class scheme (coarse-fine) also included coarse algae (CA) images in the coarse class.

2.1.3. Creation of Training Dataset

Both conventional and deep convolutional machine learning models require input data with assigned labels to learn (i.e., train) patterns and relationships within the data that produce the desired label. Additional labeled data are essential to assess the accuracy of

model predictions for data not used in the training process (i.e., validation). The accuracy of an image classification model depends in part on the quantity and quality of training images and associated class labels. Large datasets reduce the likelihood of model overfitting, which occurs when the model essentially memorizes the training data rather than identifying the underlying patterns that predict a class. For instance, using training data where all pebble images were drawn from the same AUV mission might "teach" the model that the specific lighting conditions present on that day are the best way to predict pebble images in general. If the model were then applied to pebble images from another mission with different lighting, the model may predict the class inaccurately. Here, we sought to assemble a training dataset large and diverse enough to represent the varied environmental and lighting conditions encountered in the Laurentian Great Lakes, while ensuring that substrate classes were approximately evenly represented in the dataset. Imbalances in the number of training images between classes can result in overprediction of the most abundant classes. Even if the AUV had encountered fine substrates 90% of the time, a well-performing model would still accurately predict the other 10% of substrate types.

A manually labeled dataset of images was created and split into training and validation datasets used in model training. Images were selected from eighteen AUV missions from Lake Michigan, representing different geographies, light conditions, and times of day (Figure 1). Images from the missions were included in the dataset if they had complete and valid metadata, were taken at altitudes ranging from 0.5 and 5 m above the lake bottom, and were clear enough that bottom features were readily identifiable. In all, a set of 7282 images were subsequently labeled for benthic substrate classification.



Figure 1. Locations of AUV images used for developing the training dataset for geologic substrate classification using machine learning models.

Images were initially labeled using the 9-class scheme (Table 2; Figure 2), plus a coarse algae class consisting of images that are clearly coarse but were more than 50% obscured by vegetation. Each image was independently labeled by three trained observers using a custom program with a graphical user interface (GUI) that included a toggle for 1-, 10-, and 100 cm grids over each image for scale. Scale was established using the pixel size calculation that accounted for lens characteristics and magnification effects caused by different indices of refraction between air and water.

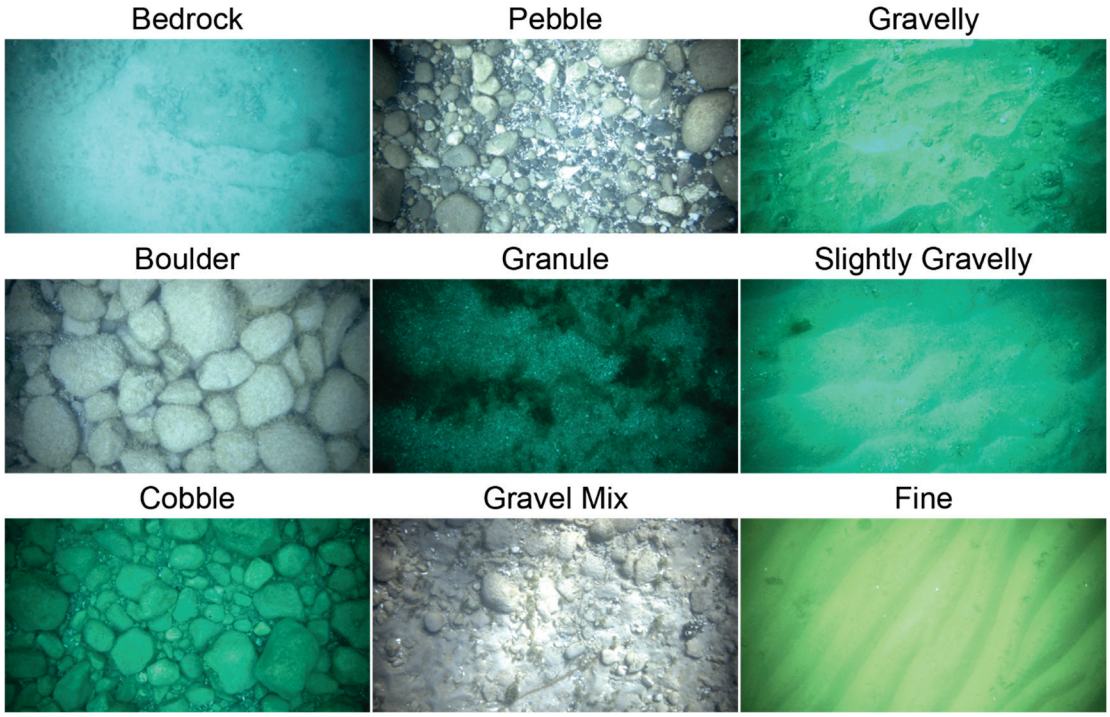


Figure 2. Example AUV images for each of the 9 classes used to manually classify images.

To calculate scale in an image, we assumed a pinhole camera model viewing a planar lakebed at a known altitude [31]. We then calculated a magnification factor, which would account for scaling due to the lens port between the camera and the water. The length in pixels of a given distance in an image could then be calculated using the following equation:

$$l = (nf/w\alpha) * L/a, \quad (1)$$

where l is the length in pixels on an image, L is the length on the lakebed in meters, n is the number of pixels on the wide length of the sensor, f is the focal length of the camera lens, w is the width of the camera sensor, α is the magnification factor, and a is the altitude of the drone. The GUI calculates the distance between grid lines based on this equation, given the altitude of each image, and overlays the grids on the image. This allowed labelers to have a relative scale for objects in an image and more accurately assign a classification for that image.

The three assigned labels for each image were compared, and a final definitive label was determined based upon decision rules (Table 4). Images with full consensus among the three labelers were assigned with the consensus label as their final class. Images where two labelers agreed on the classification but the third did not were assigned the majority classification of the two labelers in agreement. For example, if two people labeled an image as boulder and the third person labeled the image as cobble, a class directly adjacent to boulder and easily confused for edge-cases, the image was assigned a final label of boulder. When there was no agreement among the three labels for an image, it was arbitrated by the senior scientist (Esselman) for a final assigned label. However, if two or more labelers

assigned the image class as "unknown" for any reason or if the arbitrator determined the image class was "unknown", then the image was discarded from the training dataset.

Table 4. Rules for determining a final class from the results of three independent expert labelers.

Condition	Final Label
All three labelers agree on classification	Three-way agreed-upon class
Two labelers agree on the classification, one labeler assigns a different classification	Two-way agreed-upon class
No labelers agree on classification	Arbitrated by senior author and assigned final class

Of the 7282 total images labeled in the complete Geisz et al. (2024) dataset [32], full consensus was reached on 1873 images (25.7% of all images). These images were assigned the agreed-upon label. Agreement was reached by two labelers for 4079 images, representing just over half (56%) of the total labeled images. These image labels were assigned the two-out-of-three consensus class label. The remaining 1330 images (18.3% of all images) had complete disagreement between the three trained labelers and were arbitrated by the senior scientist, who determined the final class assignment.

Although the AUV was programmed to travel 1.75 m above the lakebed, the actual altitude varied because of the onboard altitude sensor. As a result, altitude varied by image. To minimize variation in the ground sample resolution (GSR) from one image to another, we subsampled a larger labeled dataset of 7282 images to include 4956 images with an altitude between 1.60 m and 2.10 m (Table A1). These thresholds were selected based on the histograms of the altitude values for all images to limit the outside influence of image scaling. The mean altitude of this subset was 1.88 $\pm$ 0.13 m. Since image altitude varied, the GSR of a pixel ranged from 0.44 mm to 0.57 mm per image, where images taken farther from the lakebed had a coarser GSR than images taken closer to the lakebed. The GSR of an image at the average altitude of 1.88 m was 0.51 mm. See Table A2 for the GSR of images by altitude.

2.2. Machine Learning Classification Models

The ML classification models used 4956 images taken from the AUV color camera taken between 1.6 m and 2.1 m from the lakebed to classify benthic images by substrate type. The labeled dataset was used to train and compare two types of ML models: (1) an RF classifier using a vector of engineered features derived from each image plus plane standard deviation derived from the stereo point cloud, and (2) a DNN classifier using the raw images only. Each of the two types of ML models were trained to classify AUV images based on the 9-class, 6-class, and 2-class schemes (Table 3). Random forest models were selected because they are very efficient at classifying tabular data without overfitting for smaller datasets, although using RF models to classify images necessitates pre-computing a series of features from each image. In contrast, neural networks were selected as a comparison because they are effective at image classification problems without pre-computing any features but can require larger amounts of data for training. Five-fold cross-validation was applied on all models to compare the RF and DNN models and test for model stability (i.e., that a single model training run was not anomalously high, low, or highly variable).

All ML models were trained using down-sampled images at 60% of the native spatial resolution to reduce the number of trainable model parameters and allow the program to run on a desktop workstation as previously described. Based on a series of experiments, training with full native resolution images was computationally prohibitive because the number of model parameters was too great, but training on excessively down-sampled images (down-sampled to less than 60% of native resolution) significantly reduced the model accuracy. Since excessive down-sampling likely was missing valuable texture information, all images were down-sampled from 4096 $\times$ 2176 pixels (native resolution) to 2458 $\times$ 1306 pixels (60% of native resolution). Previous research comparing ML model accuracy and performance utilized k-fold cross-validation because it reduced the likelihood

of model overfitting to a single subset of training data, improved model generalization, and facilitated objective comparison of different types of ML models [33,34].

To explore issues of scale and the importance of the image altitude from the lakebed, two variations of RF and DNN models were trained. One set of RF and DNN models was trained without including image altitude as a model input, while the second set of RF and DNN models did include this value explicitly for every image.

2.2.1. Random Forest (RF) Models

RF are a type of ensemble model that predict class membership based on the most common label predicted from many individual decision trees, usually several hundred to thousands [35]. Each tree was trained on a random sample of the labeled image dataset. The first step in RF model development was to create a feature vector of various metrics that were heuristically expected to correlate with the substrate type described below and summarized in Table 5. The features were calculated for each image. This "feature engineering" step potentially improves the interpretability of the model by providing control over the model inputs. In addition, the feature vector requires far less memory to train on than whole images, which allowed for faster training and prediction once the features were calculated. However, computing the features was time consuming.

Table 5. Features derived for each image used in the local classification of substrate type. With 16 Local Binary Pattern metrics and 4 FFT annuli, the final feature vector included 30 elements to be predicted by the RF model.

Feature Vector Index	Metric
1	Intensity Variance
2	Edgeness
3	GLCM Contrast
4	GLCM Dissimilarity
5	GLCM Homogeneity
6	GLCM ASM
7	GLCM Energy
8	GLCM Correlation
9–24	LBP 1–16
25	FFT Norm
26–29	FFT Annulus Norms
30	Plane Standard Deviation

In addition to downscaling, the images were converted to grayscale and their histograms were equalized. The histogram equalization and grayscale conversion were implemented to prevent the model from overfitting due to the hue and lighting conditions that may be similar in certain AUV missions but may not transfer to other data, making the model less robust. Any metrics calculated from the images had these preprocessing steps implemented prior to calculation. Both preprocessing and metric calculation was conducted using Python scripts, which rely on the numpy [36], OpenCV [37], pandas [38], and scikit-image [39] libraries.

- **Intensity Variance**: The brightness, or intensity, of a pixel in a grayscale image is simply an 8-bit integer, a number between 0 to 256. The variance of these values indicates how much these numbers are dispersed about their mean. Heuristically, we would expect to see more variation where there are more shadows and reflective surfaces.
- **Edgeness:** We would expect images with more "things" in them—usually rocks or shells or plants—would have more edges. We apply a canny edge detection algorithm [40] with sigma set to 3 to the image and calculate the proportion of pixels in the image that are considered edges. This metric we call "Edgeness".
- **Gray Level Co-Occurrence Matrix:** Gray Level Co-Occurrence Matrices (GLCMs) are often used for analyzing texture in images. Four GLCMs were calculated for each

pre-processed image, using an offset of 1 and angles of 0, $\pi/4$, $\pi/2$, and $3\pi/4$. The contrast, dissimilarity, ASM, energy, and correlation, all defined in paper [41], were calculated from these matrices and averaged over the four angles. Each metric was a separate feature in the feature vector.

- **Local Binary Patterns:** Local Binary Patterns (LBPs), first defined in [42], are commonly used for texture analysis. Using only 4 neighbors 1 pixel away, a histogram of the number of pixels falling into each of the 16 local binary pattern bins in the image was obtained. The number of pixels in each bin became a metric in the feature vector, resulting in 16 features.
- **Fourier Metrics:** The discrete 2D Fast Fourier Transform (FFT) of each image was taken [43]. Then, the Frobenius Norm was taken, first of the entire FFT matrix, then of the FFT matrix masked to highlight certain frequencies in the image. In total, 4 annuli-shaped masks were used so that the angle of the frequency was ignored. With the minimum side length of the images being 1306 pixels, the 4 masks were from 0 to 326 pixels, 326 to 653 pixels, 653 to 979 pixels, and 979 to 1306 pixels. The norms for each of the different matrices each gave a metric for the feature vector, totaling 5 feature metrics.
- **Point Cloud Standard Deviation:** In addition to color images, the AUV collects stereo imagery from two 2 Mp grayscale cameras and calculates disparity maps and point clouds at four frames per second (the cameras are not synchronized so there may have been some offset to the images). The roughness of the point clouds was assumed to correlate to the roughness of the benthic surface in the color image. We calculated the standard deviation of the heights of the points as a measure of roughness, where height is defined to be the distance from the plane of best fit through the point cloud. The plane of best fit was identified using principal components analysis (PCA) by taking the first and second principal components (PC1 and PC2) to represent lateral dimensions of the data. The third principal component (PC3) is orthogonal to the lateral plane and should capture vertical dispersion in the data. This metric was calculated for each point cloud in a mission and treated as a time series. Each color image was assigned a plane standard deviation value by linearly interpolating the plane standard deviation time series to the timestamp of the image.

Random forest modeling was conducted using the scikit-learn Python library. Each forest was composed of 100 decision trees, using the default parameters defined by the Python library. Five-fold cross-validation was used for RF model development to reduce the disproportionate influence of one training–validation split. Five-fold cross-validation uses a different randomly selected 80% of the dataset for model training and the remaining 20% applied as a test set to measure model performance. This process was repeated five times without replacement, resulting in five different RF models, where each one was trained and evaluated on a different 80% and 20% of the data, respectively. A final model is trained on the entire dataset.

2.2.2. Deep Neural Network (DNN) Models

The 9-, 6-, and 2-class DNN image classifiers were developed in Python using Tensorflow 2.09 [44] and Keras [45]. Each model consisted of a data normalization layer, multiple random image augmentation layers (i.e., hue adjustment, brightness adjustment, contrast adjustment, image rotation, and horizontal and vertical image flip), five 2D convolutional sequences (i.e., 2D convolution and 2D max pooling), two dense layers with dropout between them, and a softmax activation layer (Figure 3).

All input images were first normalized from 8-bit color depth (0 to 255) to 0 to 1. Data normalization was important because it equalized the input data ranges such that each model was more stable and avoided exploding gradients. This normalization also improved model transferability to images without 8-bit color depth by adjusting the normalization equation denominator to equal the maximum value for the color depth. Although the

dataset used here consisted of 8-bit color images, the models could be easily adapted to 16-bit color images.

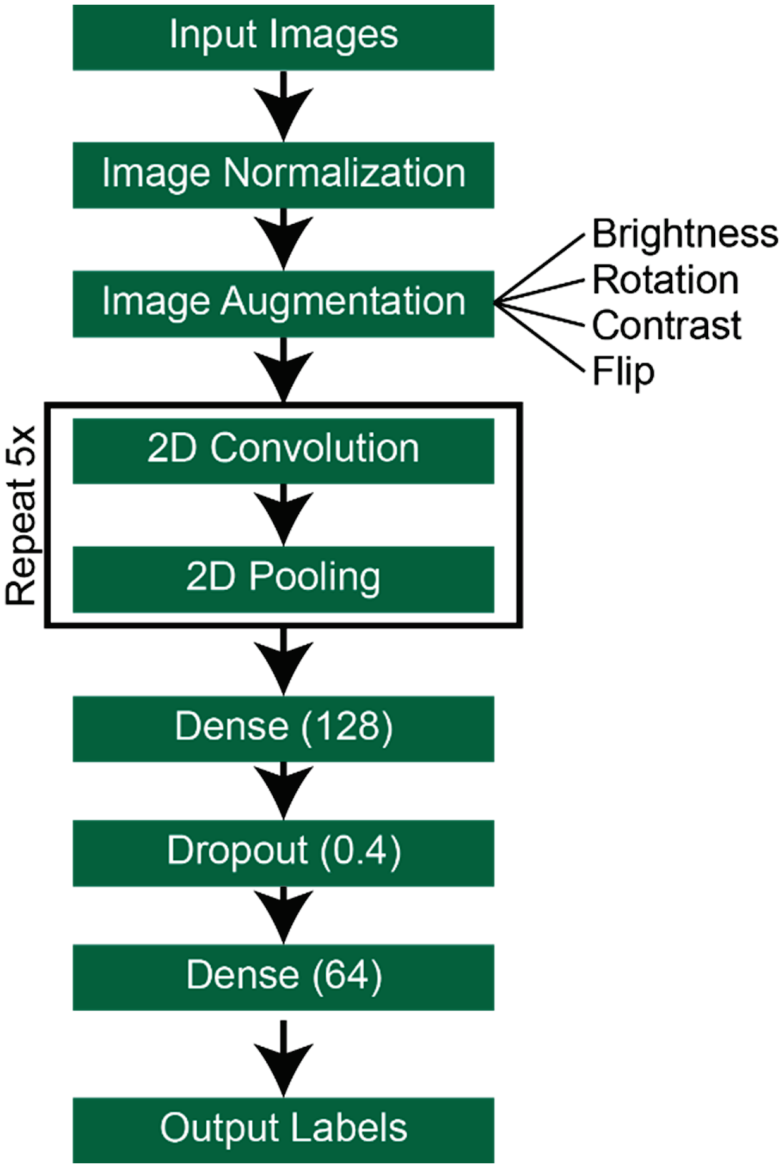


Figure 3. DNN model architecture used for training 9-, 6-, and 2-class models.

Image augmentation is an important step in developing generalizable ML models [25,33,34] and was included at the beginning of the model. Here, it consisted of a random hue adjustment, random brightness adjustment, random contrast adjustment, random image rotation, and random horizontal and/or vertical flipping. While all these augmentation steps were implemented to improve model performance and transferability, random hue, brightness, and contrast adjustments were especially important in context of the AUV image data because of the variation in natural lighting, imaging altitude, water turbidity, and camera parameters. Taken together, these factors can cause images from the same AUV transect to appear more green, blue, or gray and/or cause the images to appear underexposed relative to other surveyed areas. Since images along a single AUV transect were more likely to have a similar class, care was taken to avoid classifying all images with a given color palette as the same class. Random image hue adjustment was also included in the image augmentation process to alleviate this potential confusion and decrease model sensitivity to color biases introduced by any single AUV survey or any

parameters. Subsequent image rotation and flipping were included to limit models from using a particular region or orientation of features to determine the image classification. Randomly augmenting the dataset improved the transferability of each model by reducing overfitting to a given set of images.

Following image augmentation, a series of five 2D convolutional layer sequences were performed. Each 2D convolutional sequence included a 2D convolution layer and 2D max pooling layer, in order, and the dimensionality of the output space (i.e., filters) increased with each successive layer. The first 2D convolution layer started with 16 filters, and the number of filters increased by a power of 2, with the final 2D convolutional layer having 256 filters. Output from the final 2D convolutional layer sequence was then flattened to a single dimension and passed to a dense layer sequence. Two dense layers were used to determine the final image classification, with a dropout layer with a 0.4 probability of dropping a given node included between the dense layers. Dropout between the 128-node and 64-node dense layers was another step used to avoid model overfitting.

The last dense layer had the same number of nodes as input classes in the dataset. For example, the final dense layer in the 9-class model included nine nodes, whereas the final dense layer in the 2-class scheme models had only two nodes. The output of this layer is a probability of belonging to each of the n-classes. Condensing these probabilities to a single class label was accomplished through an activation layer using a Softmax activation function. The sparse categorical cross-entropy loss function was used for all DNN models, and learning rate scheduling and early stopping training callbacks were used to maximize the likelihood that the model found the best possible solution (i.e., finding a global minimum and maximum validation loss accuracy, respectively).

Prior to full model development, several benchmarks were tested to determine whether simpler model architectures with fewer layers or nodes per layer and/or not including one or more dropout layers would improve model performance. Changing the number of 2D convolutional sequences beyond five had no effect on model performance, aside from a substantial increase in the number of trainable model parameters and a longer training time. Decreasing the number of 2D layer sequences did, however, decrease model performance, as measured by validation training accuracy. In addition, two dense layers with a single dropout layer between them optimized model performance without overfitting or significantly increasing the number of trainable parameters. Due to the model architecture tests, the DNN models employed here included five 2D convolutional sequences and two dense layers with dropout.

Similar to RF model development, 5-fold cross-validation was applied during DNN model development. Gómez-Ríos and others used a similar approach to evaluate different convolutional neural networks (CNNs) for classifying underwater images of different coral species [34]. Final DNN model accuracy was based on a combination of all five trained models. Since training accuracy, training loss, validation accuracy, and validation loss were all logged during the training process of each model, it was possible to identify the epoch when a model began overfitting to the training data. Two training callbacks were used during model training: early stopping and learning rate scheduler. These were used during the model training process to monitor the validation loss and protect against model overfitting. Learning rate was scheduled to halve if the validation loss did not improve after 7 epochs, and this process would repeat as long as the model was training. Under the early stopping callback, model training was stopped if the validation loss did not improve after 20 epochs, and then the model weights from the best training epoch were restored.

3. Results

Preliminary comparison of the RF and DNN models with and without image altitude suggest that RF models were the same when image altitude was included, while DNN models were significantly less accurate when they included image altitude (Table A3). As a result, the remainder of this manuscript will focus on RF and DNN models that did

not utilize image altitude as a model input and were trained on the Geisz et al. (2024) dataset [32] cropped to images acquired between 1.6 m and 2.1 m above the lakebed.

Despite aiming for a dataset with a balanced class representation, the number of images in each class were not equal in the final annotated training dataset (Table 3). Boulder, cobble, and fine images were over-represented with over 700 images per class, while bedrock and granule were substantially under-represented with less than 25 images per class. The 6-class scheme had an improved class balance, although consolidated and mixed classes remained severely under-represented. A simple 2-class binary classification scheme was the most balanced, and although there was some imbalance between the coarse (3497 images) and fine (1438 images) classes, the dataset was sufficiently large to overcome this imbalance.

Five-fold cross-validation was applied for each of the three classification schemes to avoid model overfitting to the sample of data used for training. The overall accuracy for each class aggregation scheme model was calculated as the average of all 5-fold model iterations (Table 6). A confusion matrix was calculated for the aggregated 5-fold models. The diagonal boxes and values represent the number of images correctly classified as the class on the vertical axis, and the off-diagonal boxes and values represent the number of images of the vertical class incorrectly classified as the horizontal class. Together, these confusion matrices provide more detailed insight into model performance by substrate class.

Table 6. Model accuracy and standard deviation over the 5 folds for the RF and DNN models without altitude, trained using AUV images acquired between 1.60 m and 2.10 m altitude from the lakebed.

	RF	**DNN**
9-class	78.1 ± 0.9%	73.1 ± 0.9%
6-class	86.3 ± 0.9%	84.1 ± 1.4%
2-class	96.2 ± 1%	96.2 ± 0.8%

The 9-class DNN model took 214,505 s to train, and the RF model only took 4094 s to train. For the 9-class DNN model training, this total time includes training five independent models for the 5-fold cross-validation as well as a final model trained on all available data. The average time to train a single 9-class DNN model was ~35,750 s. Additional tests were conducted to determine how efficient each of these two 9-class models was when applied to an independent dataset of 1000 randomly selected AUV images. The 9-class DNN and RF models took 95 and 460 s, respectively, to classify the new images. For the RF model, approximately 455.5 s was required for feature engineering and only <0.05 s was required to classify the images.

3.1. RF Image Classification Models

Model accuracy was inversely related to the number of classes modelled, and the average accuracy for the 9-, 6-, and 2-class RF models was 78.1 ± 0.9%, 86.3 ± 0.9%, and 96.2 ± 1.0%, respectively (Table 6). RF confusion matrices (Figure 4) indicate that all RF models very accurately classified fine images, with a classification accuracy of 93.4% for fine images in the 2-class model. Other classes were more challenging to predict, as evident by the 9-class confusion matrices (Figure 4a,b). A perfect model would have values of 100% in the diagonal and 0% in all non-diagonal boxes. The 9-class RF model predicted boulder images accurately (84.8%), followed by cobbles (73.4%) and gravel mixes (52.2%). The remaining six classes of images were predicted with relatively poor accuracy, all below 30%. The greatest misclassification for the 9- and 6-class models was their tendency to misclassify many under-represented classes as belonging to the fine class, as evident by the greater numbers and darker colors along the far-right column in Figure 4a,b. Another misclassification challenge is the confusion between boulder and cobble images.

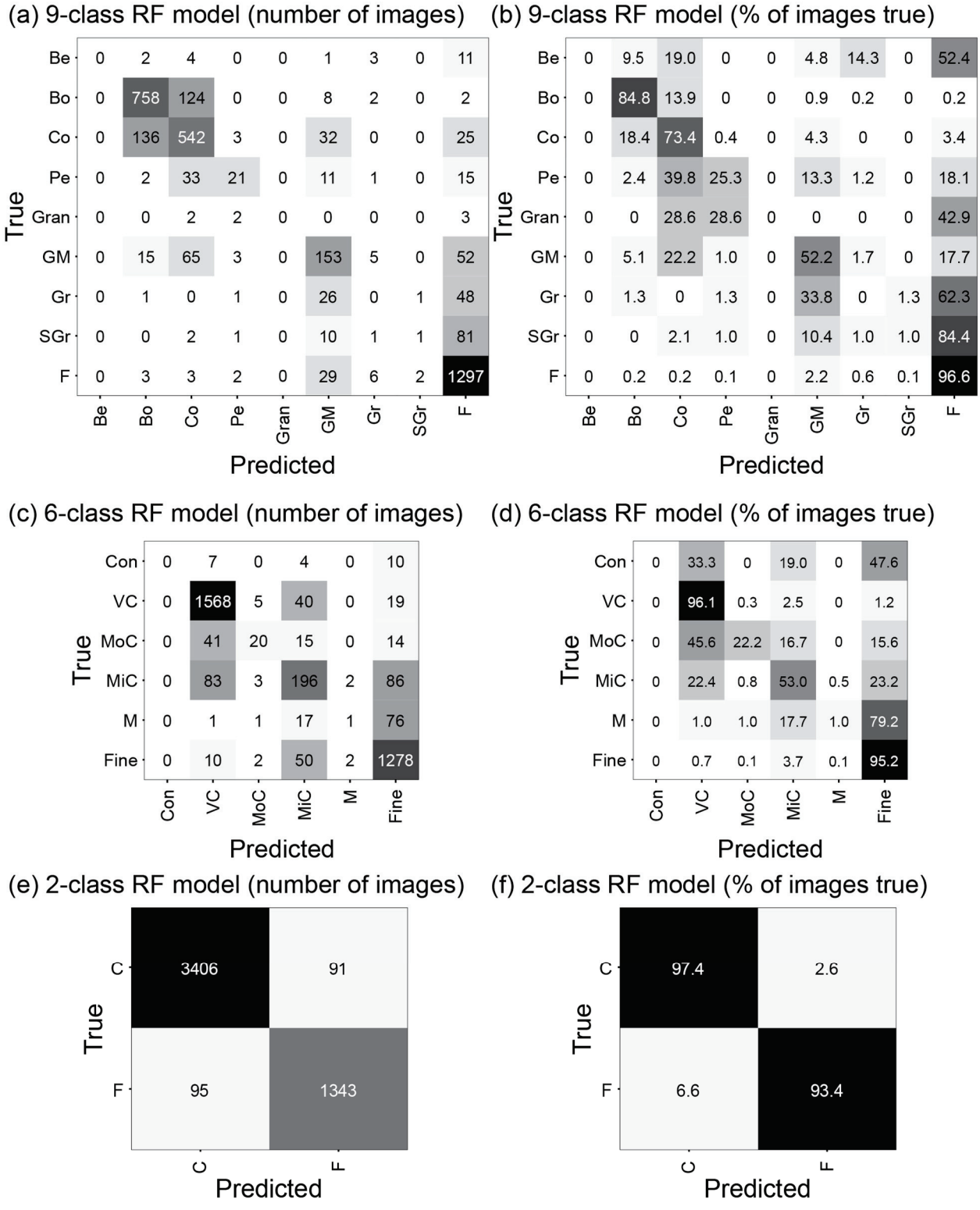


Figure 4. Confusion matrices for RF models with (**a**,**c**,**e**) number of images predicted in each class versus the true class and (**b**,**d**,**f**) percentage of images in each true class predicted as a class in the x-axis. Plots a and b represent confusion matrices for the 9-class model, plots c and d represent the 6-class model, and plots e and f represent the 2-class model. The 6-class model included consolidated (Con), very coarse unconsolidated (VC), moderately coarse unconsolidated (MoC), mixed coarse unconsolidated (MiC), mixed unconsolidated (M), and fine unconsolidated (F) classes. Acronyms for the 9-class model are shown in Tables 2 and 3.

The 6-class RF model predicted very coarse and fine images with 96.1% and 95.2% accuracy, respectively, with the next most accurate class being mixed coarse at 53.0%. Combining the boulder and cobble classes from the 9-class to the 6-class scheme appeared to reduce the amount of misclassification and improved model performance because the model did not have to differentiate between two similar classes. Like the 9-class model, the 6-class RF model tended to misclassify some under-represented classes as belonging to the fine class (see far right column in Figure 4d).

Classification with the 2-class scheme yielded the most accurate RF model, with 96.2% of images accurately classified. A total of 97.4% and 93.4% of coarse and fine images, respectively, were predicted correctly. While neither of the 9- or 6-class models classified any images as bedrock or consolidated classes, the 2-class RF model did not include any bedrock/consolidated images and did not exhibit the same misclassification issue as the 9- and 6-class models. It is important to note that the 2-class model included the additional 1405 coarse algae images as representatives of the coarse class.

3.2. DNN Image Classification Models

DNN models were successfully trained on down-sampled images with the 9-class, 6-class, and 2-class schemes. The number of trainable parameters did not vary substantially between the DNN models, although the 9-class model did have the most parameters (467,113) and the 2-class had the fewest (466,658). However, a difference of only 455 parameters was negligible and had no observable effect on model training time or performance. Model weights of the best training epoch were automatically restored for each trained model with the early stopping callback. Since validation loss was monitored and model training ceased if it did not improve over 20 consecutive epochs, none of the DNN models required all 150 training epochs available to arrive at a solution.

All training and validation training history plots tracked tightly within a given classification scheme (Figure 5), although the number of training epochs in each 5-fold model did vary. Since the number of training epochs was determined by the early stopping and learning rate scheduler criteria relative to the validation loss training curve, some models converged more efficiently than other models, resulting in minor variations in plotted training loss and accuracy curves. Training and validation loss and accuracy followed each other very closely for all DNN model realizations.

The 9-class model had the highest validation loss and lowest validation accuracy of the classification schemes tested here, followed by the 6-class model (Figure 5). The 2-class model performed the best, with the overall lowest validation loss and greatest validation accuracy. While comparing the validation accuracy metrics across the different DNN models provided insight about the general performance of a model, inspection of the confusion matrices for each DNN model provided greater insight about how well a given model predicted each substrate class.

Confusion matrix plots of the DNN models (Figure 6) help illustrate the sources of inaccuracy in each model. The 9-class DNN model accurately predicted images belonging to the fine substrate class (accuracy: 96.1%) but had very poor prediction accuracy with images belonging to the bedrock, pebble, granule, gravel mix, gravelly, and slightly gravelly classes (Figure 6a). All under-represented classes had a classification accuracy of less than 3%. Gravel mix images had a low prediction accuracy of only 23.5% and were most often conflated with pebble and granule image classes. Boulder and cobble images were frequently misclassified between each other, although the boulder image classification accuracy (80.4%) was greater than the cobble image classification accuracy (64.7%).

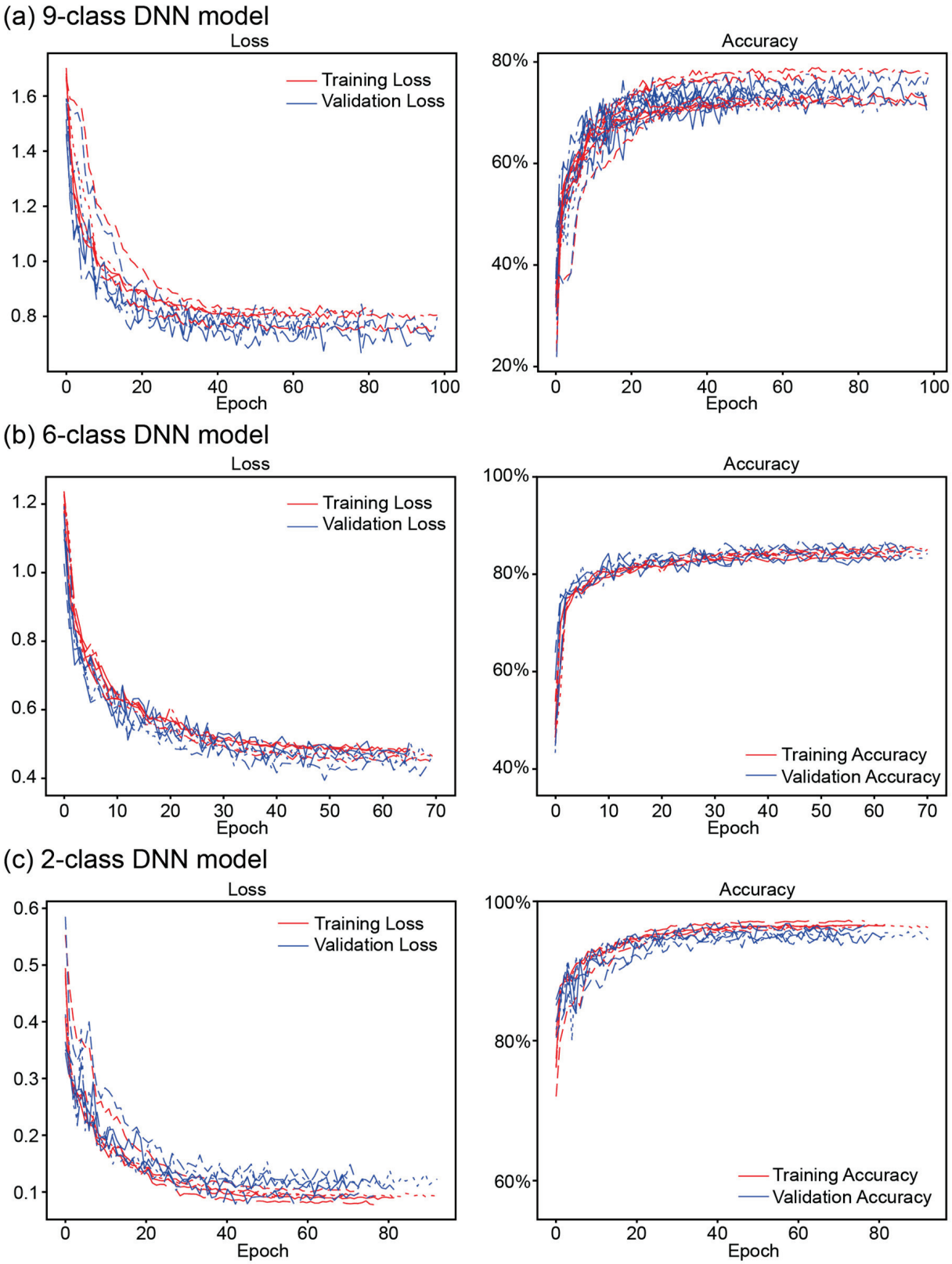


Figure 5. DNN mode loss and accuracy training plots for (**a**) 9-class, (**b**) 6-class, and (**c**) 2-class schemes. Red lines represent the training loss and accuracy, and the blue lines represent the loss and accuracy using images withheld for model validation (i.e., validation loss and validation accuracy). Since each model was trained and validated using 5-fold cross-validation, each k-fold iteration is represented its own line type on the plots.

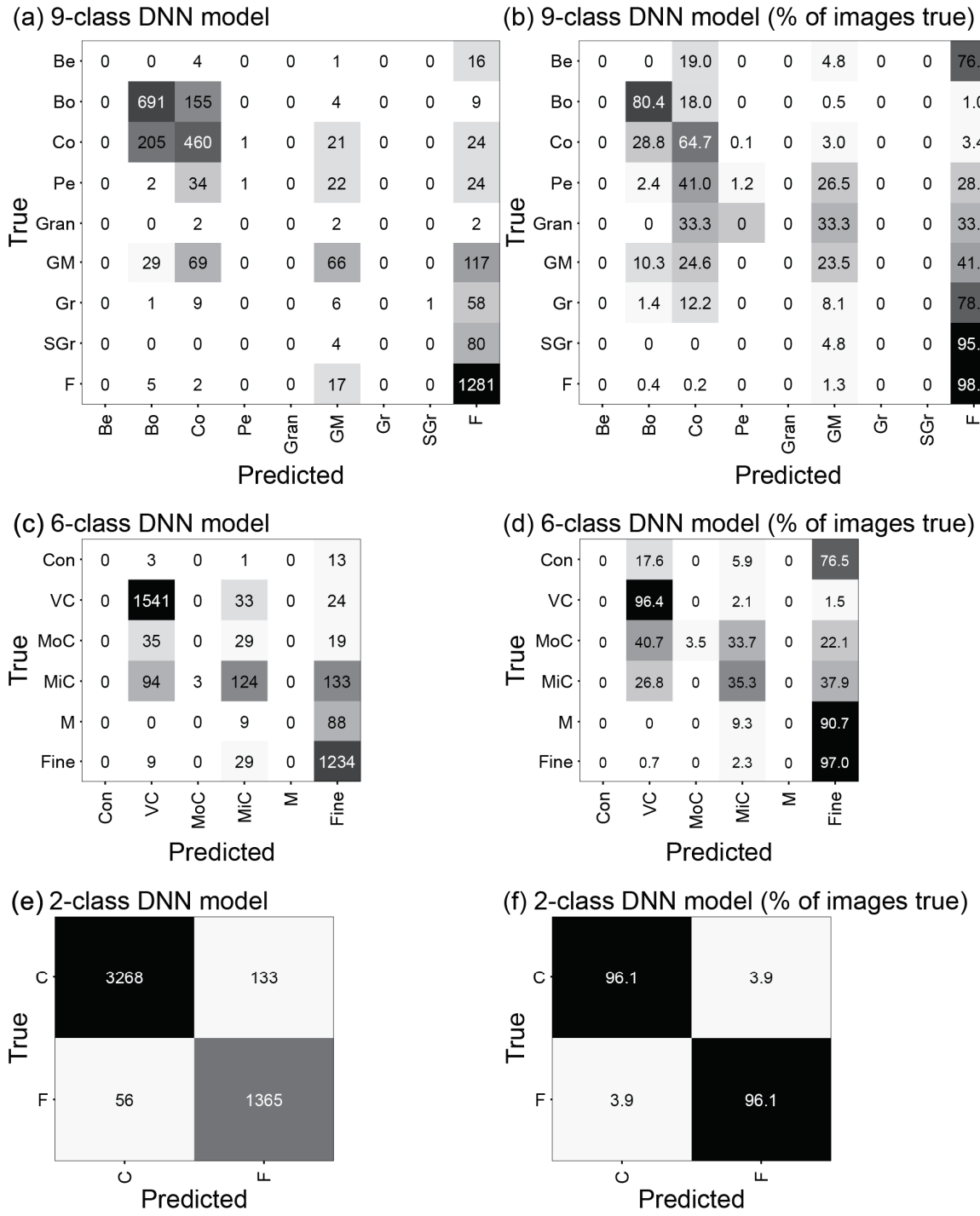


Figure 6. Confusion matrices for (**a**,**b**) 9-class, (**c**,**d**) 6-class, and (**e**,**f**) 2-class DNN models. Plots a, c, and e represent the number of images predicted as a given class versus their true class assignment, and plots b, d, and f represent the percentages of images labeled as the true class and predicted as the class on the x-axis.

The 6-class model also accurately predicted images belonging to the fine class (97.0%). A significant improvement in the 6-class over the 9-class scheme is its ability to accurately classify 6-class aggregated classes better than the individual 9-class scheme. Aggregating boulder and cobble to the 6-class label very coarse unconsolidated (VC) resulted in a classification accuracy of 96.4%. Mixed coarse unconsolidated (MiC) images, composed of gravel mix and gravelly classes, were classified more accurately (35.3%) than either of

the two 9-classes that compose this 6-class category. The 6-class model still struggled to accurately moderate coarse (MoC) and MiC and tended to confuse these two categories with images from the fine category.

The 2-class DNN model had the best classification accuracy for coarse and fine classes, with a classification accuracy of 96.1% for both coarse and fine images. Only 3.9% of the labeled fine images were misclassified as coarse, and the same percentage of coarse images were misclassified as fine.

4. Discussion

4.1. Manual Image Labeling

The results demonstrate that RF and DNN models can reasonably predict the dominant substrate classes present in whole images collected at altitudes suitable for AUV operations. Given the large numbers of images that AUVs can collect (i.e., ~63,000 per 3.5 h deployment for our AUV), automated image classification presents one of the only means to fully analyze such data. Compared to the alternative of manual classification, the automated classification of images offers specific advantages for scalability, objectivity, and transferability.

The ML models presented here are substantially more scalable than manual image labeling. Manual labeling can be reasonably accomplished for relatively small datasets of up to a few thousand images. However, manual labeling becomes prohibitively expensive or impossible for datasets composed of tens of thousands of images or more, and ML models are well poised to eliminate these issues. Whereas an expert may take 10 s or more to accurately determine the substrate class in an image, a well-validated ML model can predict the substrate class within a fraction of that time. As a result, classifying the substrate present in an image becomes more advantageous as the number of images required for labeling continues to grow.

ML models can be less subjective compared to manual image labeling [46]. It is possible for a single person or group of people to classify many images, in which case the ML model trained on these data would likely capture subtle biases or errors from the sole person annotating the training data. However, ML models trained on a dataset generated by multiple people may attenuate this subjectivity by leveraging the knowledge imparted by the different people generating the labels. In this way, the ML models would be buffered from biases or errors introduced by any single person annotating the training data. Both human labelers and ML models will tend to assign a single class label to a new image where class membership is ambiguous (i.e., edge cases) instead of classifying it as "unknown." The accuracy of any ML model is at least partially determined by the accuracy and consistency of the training data. While it is possible for an ML model to misclassify images, a well-developed ML model can more consistently classify images compared to manual image classification.

ML models may allow for greater transferability of algorithms to new geographies and datasets. In cases where the underlying surface geology, image characteristics, classes present, or geography have changed, it may be necessary to re-train the ML model, although this re-training may be more efficient than a person re-learning with the new dataset. For instance, the ML models presented here were trained exclusively on images from Lake Michigan, which shares some similarities to the surface geology of Lake Huron and Lake Erie but different surface geology than Lake Superior and Lake Ontario [47,48]. Image classification within a single dataset for a single person can be relatively simple when that individual has adequate knowledge of the local geology and physical processes; however, applying the same set of criteria to a new dataset representing a different time or geography can necessitate an extensive re-learning process where drift can also be introduced, as previously discussed. Image classification with ML models may be more transferable to a new dataset and geographies, even if some model re-training is required.

4.2. RF Image Classification Models

The relatively high number of images represented by boxes along the diagonal in all RF model confusion matrices demonstrate that although the model does misclassify some images, misclassified images were generally predicted as belonging to a closely related class. This is consistent with previous work, where the misclassification of cobble and boulder images presented a challenge [25]. Model accuracy was inversely related to the number of classes, which suggests that RF models were able to capture inter-class variability well enough to reasonably distinguish between classes, but additional criteria may improve models with more classes.

The RF model results presented here suggest that misclassification issues may be at least partially addressed by class aggregation. The 9-class RF model misclassified 13.9% of the labeled boulder images as cobble and misclassified 18.4% of the labeled cobble images as boulder. However, when these two classes were aggregated to a combined very coarse class, as in the 6-class scheme, the classification accuracy for the aggregated class improved to 96.1%. Further improvements in classification accuracy for both classes in the 2-class scheme support aggregation as one possible approach to improving ML model performance by reducing ambiguity between classes. Although having a high degree of specificity in classes can be beneficial depending on the application, aggregating adjacent classes can substantially reduce misclassification by ML models.

Among the 29 total features engineered for RF models, the planar standard deviation was the most influential for all classification schemes. While this influence was disproportionate for the 9-class RF model, its influence was closely followed by three additional LBP metrics in the 6-class RF model (LBP 6, LBP 11, and LBP 13). The same four features (plane standard deviation, LBP 6, LBP 11, and LBP 13) were most influential for the 2-class RF model, although the plane standard deviation and LBP 6 were slightly more influential than the other LBP features. These feature importance rankings suggest that 3D information, such as plane standard deviation, may be particularly helpful for distinguishing between classes in the RF models. We conjecture that the LBP metrics become more important in the 2-class model due to the introduction of the coarse algae images, and the model uses the texture of the algae as a predictive factor.

4.3. DNN Image Classification Models

The 9-class DNN model had a lower accuracy than the 6- and 2-class DNN models. It is unlikely that this discrepancy in DNN model performance was caused by model architecture or hyperparameters since all DNN models had the same model architecture and hyperparameters. Rather, differences in model performance were more likely a result of the 6- and 2-class schemes being simpler with clearer distinction between the classes. As discussed with the RF models, class aggregation can enhance classification by pushing some of the inter-class variability into a single class's variability.

The DNN models did not appear to overfit the training data for any of the 5-fold models in the different classification schemes. Training and validation lines tracked very well with each other in Figure 5, suggesting that the model architecture and data augmentation steps employed here were effective to develop a robust 9-, 6-, and 2-class DNN model.

Developing and training robust ML models is predicated on employing appropriate model architecture and augmentation processes. Previous research demonstrates that incorporating image augmentation and random dropout layers can effectively improve ML model robustness [19,33]. Prior to settling on the presented DNN model architecture, multiple architectures were tested, including some without image augmentation, different number and order of image augmentation processes, and/or dropout layers. Different training, validation, and testing splits were also tested to determine what architecture, augmentation scheme, and training split produced the most generalizable model based on an independent evaluation data split. None of the DNN models exhibited overfitting (Figure 5) because of the architecture experiments and care taken during model development. Each set of training and validation loss and accuracy lines in Figure 5 track well

and do not substantially deviate from each other for any given 5-fold model realization. Employing learning rate scheduling and early stopping criteria further improved model performance and generalization by limiting model overfitting and restoring the model at the epoch with the lowest validation loss. Model loss and accuracy were relatively similar across the different 5-fold model realizations for all DNN models, highlighting the stability and robustness of the models to any given data.

4.4. Model Comparison

Overall, the RF and DNN models performed comparably for each classification scheme. Using a 5-fold cross-validation approach for both types of ML models decreased the likelihood of model overfitting, improved model generalization, and facilitated comparison of the RF and DNN models. The 9-class models were less accurate than the 6- and 2-class models for both model types, likely because of simpler classification schemes, more equitable class representation, and less sensitivity to image scaling issues with the 6- and 2-class models.

Multi-class classification becomes increasingly challenging as the number of classes increases because the visual characteristics used to distinguish any additional classes become more nuanced and subtle. Reif and others found a similar pattern using support vector machines (SVMs) to classify benthic habitat in southwestern Lake Michigan from a combination of in situ images and videos, satellite, and airborne sensors [5]. Their ML models were more accurate for Tier II classification with two classes compared to their Tier I classification models with three classes. Classification using a scheme with fewer classes was more accurate than classification with more classes. In the current paper, both RF and DNN models exhibited an inverse relationship between the number of classes and model accuracy. Ternon and others also noted a similar pattern when mapping temperate rocky reefs using underwater photogrammetry and proposed a strategy to mitigate this issue by aggregating neighboring classes in the context of ecological function and context [25]. Our aggregation scheme from the 9-class to 6-class and again from 6-class to 2-class and the associated improvements in model accuracy with decreasing number of classes demonstrate that aggregating functionally similar or equivalent substrate classes is an effective strategy to improve ML model accuracy and generalization.

Much of the misclassification for both RF and DNN models occurred between adjacent classes, most notably misclassifying boulder and cobble images in the 9- and 6-class models. Previous research highlights the challenge of accurately classifying boulder to pebble size substrates, noting a substantial misclassification of substrates with cobble and boulder size particles [25], possibly due to the lack of explicit image or data scale information. By extension, the simpler classification scheme of the 2-class scheme allowed the RF and DNN models to better solve a binary classification problem where images are either one substrate type or another. The two classes are also more visually distinct from each other, where coarse images tend to be more heterogenous and have more edges than a smoother fine image. Building on previous research, we support the proposal that simplification of the classification scheme, where possible, can improve ML model accuracy and generalizability.

One limitation of the ML models in this paper is their inability to directly account for differences in the altitudes at which images were collected and the resulting variation in GSR and the area captured within each image. The impact of scale and varying image footprint size is a potential reason the 9-class models struggled to classify boulder, cobble, and pebble images accurately and consistently. For instance, both 9-class models had challenges resolving the boulder class from the cobble class, possibly because subtle changes in the AUV altitude from one image to the next caused boulders in one image taken farther from the lakebed to appear similar in size to cobbles in another image taken closer to the lakebed. The program used to annotate images during the development of the training dataset did include an optional grid overlay corresponding to 0.01-, 0.1-, and 1.0 m cells, which provided valuable image scale information that was useful when accurately differentiating boulder and cobble images. However, when AUV altitude was explicitly

included in the RF or DNN models, none of the models improved significantly. The DNN models with altitude even displayed a significant decrease in accuracy compared to DNN models without altitude. Cropping the dataset to include only AUV images acquired between 1.60 m and 2.10 m altitude above the lakebed substantially reduced the altitudinal variation and variability in GSR between images, thereby minimizing the direct effect of varying image altitude on the models and classification results.

Class imbalance in training data represents a challenge for developing generalizable ML models [34] and is another reason the 6-class and 2-class models may have performed better than the 9-class model. RF and DNN 9-class models performed poorly at predicting the under-represented classes in the training dataset: bedrock, pebble, granule, gravel mix, gravelly, and slightly gravelly. Under-represented classes presented a challenge to developing accurate ML models because the models were less likely to encounter an image of these classes during the training process. Since they were less likely to "see" an image from under-represented classes, the models may have had a difficult time learning how to distinguish these from adjacent classes. In contrast, the 9-class models did accurately predict fine class images because fine images were well represented in the training data and tended to be more homogenous in appearance with fewer hard edges visible in the image. This is not a unique challenge and is well documented by previous coral reef mapping and classification literature [22,34,49,50], where poor representation of one or more classes can disrupt the model's ability to accurately classify those classes/objects. While under- and over-sampling may help mitigate class imbalance effects, how the data are under- or over-sampled can significantly affect the model and its robustness to new data [51]. Ultimately, a better long-term solution to class imbalance is to cultivate more balanced training data and improve the likelihood that a ML model "sees" all classes during the training and evaluation processes.

The most notable difference between RF and DNN model development was the feature engineering conducted for the data fed into the RF model. The DNN model essentially engineers or learns these features through the series of 2D convolutional layers in the model. The feature engineering step took 4073 s but allowed the RF models to train 8.7 times faster (21 s for model training) than DNN models (35,750 s for model training), but the feature estimation preprocessing required substantial time and resources. In contrast, predictions can be made from the trained DNN (95 s to classify 1000 images; 0.09 s per image) more quickly than the RF (460 s to classify 1000 images; 0.46 s per image), because the DNN does not require the pre-calculation of engineered feature values, which took almost the entire RF model prediction time. Considering the aggregate time required to prepare for, train, and apply both model types, the DNN is better suited for application to large datasets. Given comparable RF and DNN model accuracies for each of the three classification schemes, the DNN model can reasonably be seen as the preferred model on the grounds of computational efficiency for applications.

4.5. Future Work

Depending on the application, the 2-class model and 6-class models may have immediate utility. The high accuracy of the 2-class models indicates that either the RF or DNN approaches can be used now to produce highly accurate predictions for data with similar altitudes and other visual characteristics (lighting, hue, etc.). Since the DNN models included image augmentation procedures within the training process, these models may be more robust to classifying new data of varying hue, lighting, and other characteristics. For the 6-class model, the main confusion was for MoC and MiC images, which are more heterogeneous than other classes. Given that boulders and cobbles in the VC class may provide similar functions in an ecological context (e.g., for fish spawning and hatching [52]), the 6-class model may have sufficient accuracy now for many ecological purposes.

Future work should: (1) continue to test model performance on images gathered from a broader geography (and improve the models if necessary), (2) explore how sensor and mission parameters affect classification accuracy, (3) explore how new information from

geophysical sensors may enhance classification, and (4) continue to explore new ML model types and architectures for image classification and/or segmentation for habitat mapping applications. In addition, collaboration among various organizations and agencies should be coordinated to develop quality datasets that can be used for model development and application across broad, diverse geographic areas such as the Laurentian Great Lakes. It is notable that the parent material and glacial processes that formed the contemporary Great Lakes lakebed are very similar to those in other parts of the North American marine coast (i.e., the Gulf of Maine). Therefore, the trained models used here may be directly applicable to similar geographies outside the Great Lakes region or be useful for transfer learning applications [9].

The issue of generalizing localized or per-image classification to an entire lake basin is not trivial, as natural geographic variability in lakebed composition can be substantial. Scaling image classification to an entire lake basin is further complicated by natural variability in lighting conditions during the various AUV missions. For projects spanning multiple years, it is likely that the sensor and/or mission parameters may also change, which can further complicate the ability to efficiently train an accurate and parsimonious model that is robust to new data. Due to natural geographic and geologic variability, a new model may be required to be trained on imagery acquired in the new lake where hue, lighting, and even substrate type can vary from the training data used here.

One limitation of using AUV images to classify substrate composition and classification is their inability to resolve more subtle differences within and between some of the CMECS groups. The most notable examples of this are the inability to resolve particles smaller than 2 mm and the general inability to differentiate between bedrock, hardpan clay, and other hard surfaces. Despite the inherent differences in the geophysical properties of sand, silt, and clay, it is also not possible to accurately resolve differences between these particles with AUV images alone. Supplementing the AUV images with geophysical information, such as backscatter from an acoustic sensor (i.e., the doppler velocity logs common to AUVs), could aid significantly in differentiating the finer fractions of the substrate. Similarly, backscatter or other geophysical data may be useful in distinguishing bedrock from hardpan clay and other hard surfaces based on the differences in their sound absorption properties. Such additional instrumentation may also provide valuable data that could be used to map the water column component of the CMECS classification scheme and, therefore, result in a more complete habitat characterization. Future work should explore how additional geophysical instrumentation may effectively enhance benthic habitat characterization and mapping [3,4,33,53].

Advances in ML technologies are likely to continue, and such improvements should be explored for image classification. Developing a relatively simple, yet intuitive workflow leveraging emerging ML technologies can increase the ability of managers to map benthic habitat more efficiently in non-invasive ways. In addition, improvements with ML techniques will likely enable us to integrate new data with past information, such as field surveys and trawl surveys, which may improve our ability to accurately predict future environmental changes and focus environmental management in areas of maximum benefit.

It is possible that the feature vectors explored here contain correlated variables. Further exploration of different feature vectors should be explored as different inputs to RF models. A variable reduction approach to the feature vector creation could be beneficial in determining the most parsimonious model that is yet robust to new images. Equally as possible is that one or more essential feature vectors were absent from the stack of features included in the RF models here. The addition of other features may improve model performance and result in more robust models for benthic habitat mapping.

While the 9-class model accuracies (78.1% for RF model and 73.1% for DNN model) may be less than desired, it is important to remember that ML models can only be as good as the data and labels used to train them [46]. Generating the training dataset for the models here required coordinating multiple domain experts, and the time required to classify the entire dataset and reconcile classification labels was substantial. Since each

expert approaches image classification with their own conceptual understanding and biases, image classification by humans may introduce human subjectivity into training dataset. Reconciling the image classification labels from all three experts highlights the challenges with developing a quality training dataset and, by extension, quality image classification models. Although the goal is to develop accurate models, it is unrealistic to expect a complex multi-class model (e.g., the 9-class and 6-class models used here) to be over 90% accurate for all classes when the three experts used to develop the current dataset only agreed completely on 25.7% of image labels. Challenges with manual image classification suggest our ability to develop automated image classification models may be limited by ambiguity in dataset development. Future work should focus on developing a more extensive image dataset to help mitigate ambiguity and subjectivity in model classification and reduce significant class imbalance issues.

One limitation of the models presented here is their exclusive focus on geologic substrates. The classification scheme used in this work was specific to geologic substrate composition and did not include biogenic substrates or the biotic and water column components. The complete CMECS scheme includes a biotic component, which previous research has integrated into another modified CMECS scheme [5]. Incorporating the biotic component should be explored further, as it can directly aid in identifying and mapping important habitat modifiers like zebra and quagga mussels and/or submerged aquatic vegetation in the Laurentian Great Lakes [24,54]. Explicitly including both abiotic and biotic components of the benthic habitat mapping process could aid researchers and managers in better understanding and managing aquatic ecosystems.

5. Conclusions

The CMECS geologic substrate classification is based on the volumetric distribution of different particle sizes within a given area and therefore requires a physical substrate sample. While the information derived from this physical sample is valuable for a variety of different geological, habitat, and engineering purposes, the collection and processing of physical samples is labor/time intensive compared to the interpretation of imagery. This paper demonstrates that automated interpretation of large volumes of images is feasible using ML with reasonable accuracy, particularly if some loss of class distinctions is acceptable. Further refinement of ML approaches to classifying geologic substrates on whole images will likely lead to further improvements in accurate assignment substrates in multi-class schemes like the 9-class scheme used here.

Machine learning models have the potential to efficiently classify large volumes of AUV images compared to manual image classification and with less bias in class assignment. The classified images can be beneficial for habitat mapping and management applications. Mapping substrate composition to binary coarse and fine classes can be done with very good accuracy, although a 6-class scheme performed relatively well. Random forest models are generally more intuitive in understanding and communicating to a lay audience because the image characteristics or features can be described individually, but DNN models may be especially valuable when these features used to differentiate classes are unclear or unknown to the user, or when computational efficiency is necessary.

Images classified using RF or DNN models can provide valuable information used to classify benthic habitat more completely with the CMECS (or similar) scheme. While the original CMECS classification scheme does include other components like water column and biotic information, the modified CMECS scheme used here provides a simpler framework that can be used to map and predict benthic habitat without ambiguity from biotic habitat modifiers. Mapping benthic habitat is essential for the adaptive management of coastal environments around the world, providing invaluable information on the current state of the benthic physical, ecological, and chemical environment. High-resolution benthic maps and images can be valuable to coastal and fisheries managers by informing decision makers about native, invasive, and nuisance benthic species' abundance and distribution. Local, state, and federal agencies as well as coastal communities can also use

high-resolution maps and benthic images for mapping and monitoring the presence and distribution of sediments and pollutants. The fusion of robotics with machine learning offers great promise for providing georeferenced local predictions of geologic substrates across large spatial domains. When merged with other technologies like swath sonar, LiDAR, or satellite remote sensing, spatially extensive AUV ground truth data can serve as a valuable input to large-scale ocean and coastal mapping.

Author Contributions: Conceptualization, J.K.G., P.C.E. and P.A.W.; methodology, J.K.G., P.C.E. and P.A.W.; software, J.K.G. and P.A.W.; validation, J.K.G., P.C.E. and P.A.W.; formal analysis, J.K.G. and P.A.W.; investigation, J.K.G., P.C.E. and P.A.W.; data curation, J.K.G., P.C.E. and P.A.W.; writing—original draft preparation, J.K.G., P.C.E. and P.A.W.; writing—review and editing, J.K.G., P.C.E. and P.A.W.; visualization, J.K.G. and P.A.W.; supervision, P.C.E.; project administration, P.C.E.; funding acquisition, P.C.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Great Lakes Restoration Initiative.

Data Availability Statement: The labeled image dataset [32] used to train the machine learning models presented in this paper can be accessed at: https://doi.org/10.5066/P9N32CV7.

Acknowledgments: The authors wish to thank Chris Roussi and Ben Hart from Michigan Tech Research Institute for the technical support on programming and data collection; Dan Buscombe for machine learning advice; Scott Nelson for assistance with data management and computation; Jennifer Morris for data management assistance; Nicholas Yeager, Luke Sayler, and Alden Tilley for data labeling support; and Anthony Arnold, Scott Dwyer, Alden Tilley, Nick Yeager, Ben Beckman, Luke Sayler, Sam Pecoraro, Glen Black, and Greg Kennedy for field support. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A

Table A1. Number of images by classification type in the complete dataset [32], the dataset cropped to include only images between 1.25 m and 3.00 m altitude from the lakebed, and the dataset cropped to include only images between 1.60 m and 2.10 m altitude from the lakebed.

	Full Dataset (All AUV Images)	**1.25–3.00 m AUV Images**	**1.60–2.10 m AUV Images**
Bedrock	46	43	21
Boulder	1488	1435	894
Cobble	1133	1096	738
Pebble	136	127	83
Granule	13	13	7
Gravel Mix	500	475	293
Gravelly	124	120	77
Slightly Gravelly	138	136	96
Fine	1768	1740	1342
Coarse Algae	1936	1901	1405
Total Images	7282	7086	4956

Table A2. Calculated ground-sample resolution (GSR) in mm for different AUV altitudes from the lakebed for different portions of the complete AUV image dataset [32]; GSR is a function of the AUV altitude and camera parameters.

		Altitude	**GSR (mm)**
Dataset	Programmed AUV altitude	1.75	0.477
Full	Minimum	0.51	0.139
	Mean	2.01	0.548
	Maximum	4.91	1.339

Table A2. *Cont.*

		Altitude	GSR (mm)
1.25–3.00 m	Minimum	1.25	0.341
	Mean	1.98	0.540
	Maximum	3.00	0.818
1.60–2.10 m	Minimum	1.60	0.436
	Mean	1.88	0.513
	Maximum	2.10	0.573

Table A3. Accuracy values for all models trained, including those on the full dataset, 1.25 m–3.00 m dataset, and 1.60 m–2.10 m dataset. A RF and DNN model were trained for the 9-, 6-, and 2-class schemas for each dataset.

Dataset	Model Type	Number of Classes	Without Altitude	With Altitude
Full	RF	9	74.8 ± 0.4%	75.3 ± 0.5%
		6	85.8 ± 0.5%	85.8 ± 0.6%
		2	96 ± 0.6%	96.1 ± 0.7%
	DNN	9	72.1 ± 3.2%	33.2 ± 1%
		6	83.8 ± 2.4%	49.4 ± 0.7%
		2	96.5 ± 0.9%	73.9 ± 1.3%
1.25–3.00 m	RF	9	75.5 ± 0.8%	75.4 ± 0.5%
		6	85.7 ± 1.1%	85.8 ± 1.1%
		2	96.2 ± 0.3%	96.1 ± 0.4%
	DNN	9	72.3 ± 3.7%	33.6 ± 0.8%
		6	84.5 ± 1.7%	48.8 ± 1.4%
		2	95.4 ± 0.7%	73.3 ± 1.1%
1.60–2.10 m	RF	9	78.1 ± 0.9%	78.2 ± 0.8%
		6	86.3 ± 0.9%	86.6 ± 1.1%
		2	96.2 ± 1%	96.2 ± 1.1%
	DNN	9	73.1 ± 0.9%	37.9 ± 1.5%
		6	84.1 ± 1.4%	46.5 ± 1.9%
		2	96.2 ± 0.8%	70.7 ± 1.7%

References

1. Valentine, P.C. *Sediment Classification and the Characterization, Identification, and Mapping of Geologic Substrates for the Glaciated Gulf of Maine Seabed and Other Terrains, Providing a Physical Framework for Ecological Research and Seabed Management*; Scientific Investigations Report; U.S. Geological Survey: Reston, VA, USA, 2019; p. 50.
2. Gibbs, A.G.; Cochrane, S.A. An Integrated Approach to Benthic Habitat Mapping Using Remote Sensing and GIS: An Example from the Hawaiian Islands. In *Remote Sensing and Geospatial Technologies for Coastal Ecosystem Assessment and Management*; Lecture Notes in Geoinformation and Cartography; Springer: Berlin/Heidelberg, Germany, 2009; pp. 211–231. ISBN 978-3-540-88182-7.
3. Lucieer, V.; Hill, N.A.; Barrett, N.S.; Nichol, S. Do Marine Substrates 'Look' and 'Sound' the Same? Supervised Classification of Multibeam Acoustic Data Using Autonomous Underwater Vehicle Images. *Estuar. Coast. Shelf Sci.* **2013**, *117*, 94–106. [CrossRef]
4. Montereale Gavazzi, G.; Kapasakali, D.A.; Kerchof, F.; Deleu, S.; Degraer, S.; Van Lancker, V. Subtidal Natural Hard Substrate Quantitative Habitat Mapping: Interlinking Underwater Acoustics and Optical Imagery with Machine Learning. *Remote Sens.* **2021**, *13*, 4608. [CrossRef]
5. Reif, M.K.; Krumwiede, B.S.; Brown, S.E.; Theuerkauf, E.J.; Harwood, J.H. Nearshore Benthic Mapping in the Great Lakes: A Multi-Agency Data Integration Approach in Southwest Lake Michigan. *Remote Sens.* **2021**, *13*, 3026. [CrossRef]
6. Mabrouk, A.; Menza, C.; Sautter, W. *Best Practices for Ground-Truthing and Accuracy Assessment of Lakebed Maps in the Great Lakes: A Case Study Offshore the Bayfield Peninsula in Lake Superior*; Springer: Berlin/Heidelberg, Germany, 2022; p. 25. [CrossRef]
7. Benoist, N.M.A.; Morris, K.J.; Bett, B.J.; Durden, J.M.; Huvenne, V.A.I.; Le Bas, T.P.; Wynn, R.B.; Ware, S.J.; Ruhl, H.A. Monitoring Mosaic Biotopes in a Marine Conservation Zone by Autonomous Underwater Vehicle. *Conserv. Biol.* **2019**, *33*, 1174–1186. [CrossRef] [PubMed]
8. Mahmood, A.; Ospina, A.G.; Bennamoun, M.; An, S.; Sohel, F.; Boussaid, F.; Hovey, R.; Fisher, R.B.; Kendrick, G.A. Automatic Hierarchical Classification of Kelps Using Deep Residual Features. *Sensors* **2020**, *20*, 447. [CrossRef]

9. Mohamed, H.; Nadaoka, K.; Nakamura, T. Semiautomated Mapping of Benthic Habitats and Seagrass Species Using a Convolutional Neural Network Framework in Shallow Water Environments. *Remote Sens.* **2020**, *12*, 4002. [CrossRef]
10. Mohamed, H.; Nadaoka, K.; Nakamura, T. Towards Benthic Habitat 3D Mapping Using Machine Learning Algorithms and Structures from Motion Photogrammetry. *Remote Sens.* **2020**, *12*, 127. [CrossRef]
11. Wentworth, C.K. A Scale of Grade and Class Terms for Clastic Sediments. *J. Geol.* **1922**, *30*, 377–392. [CrossRef]
12. Trefethen, J.M. Classification of Sediments. *Am. J. Sci.* **1950**, *248*, 55–62. [CrossRef]
13. Schlee, J.S. *Atlantic Continental Shelf and Slope of the United States: Sediment Texture of the Northeastern Part*; Professional Paper; US Geological Survey: Seattle, WA, USA, 1973.
14. Shepard, F.P. Nomenclature Based on Sand-Silt-Clay Ratios. *J. Sediment. Res.* **1954**, *24*, 151–158.
15. Folk, R.L. *Petrology of Sedimentary Rocks*; Hemphill Publishing Company: Austin, TX, USA, 1980.
16. United States. National Ocean Service and United States. Federal Geographic Data Committee. Coastal and Marine Ecological Classification Standard (CMECS). 2012. Available online: https://repository.library.noaa.gov/view/noaa/27552 (accessed on 6 June 2022).
17. Harter, S.L.; Paxton, A.B.; Winship, A.J.; Hile, S.D.; Taylor, J.C.; Poti, M.; Menza, C. *Workshop Report for Approaches to Mapping, Ground-Truthing, and Predictive Habitat Modeling of the Distribution and Abundance of Mesophotic and Deep Benthic Communities*; National Oceanic and Atmospheric Administration: Silver Spring, MD, USA, 2022; p. 38.
18. Burns, C.; Bollard, B.; Narayanan, A. Machine-Learning for Mapping and Monitoring Shallow Coral Reef Habitats. *Remote Sens.* **2022**, *14*, 2666. [CrossRef]
19. Chen, X.; Hassan, M.A.; Fu, X. Convolutional Neural Networks for Image-Based Sediment Detection Applied to a Large Terrestrial and Airborne Dataset. *Earth Surf. Dynam.* **2022**, *10*, 349–366. [CrossRef]
20. González-Rivero, M.; Beijbom, O.; Rodriguez-Ramirez, A.; Bryant, D.E.P.; Ganase, A.; Gonzalez-Marrero, Y.; Herrera-Reveles, A.; Kennedy, E.V.; Kim, C.J.S.; Lopez-Marcano, S.; et al. Monitoring of Coral Reefs Using Artificial Intelligence: A Feasible and Cost-Effective Approach. *Remote Sens.* **2020**, *12*, 489. [CrossRef]
21. Pavoni, G.; Corsini, M.; Pedersen, N.; Petrovic, V.; Cignoni, P. Challenges in the Deep Learning-Based Semantic Segmentation of Benthic Communities from Ortho-Images. *Appl. Geomat.* **2021**, *13*, 131–146. [CrossRef]
22. Raphael, A.; Dubinsky, Z.; Iluz, D.; Netanyahu, N.S. Neural Network Recognition of Marine Benthos and Corals. *Diversity* **2020**, *12*, 29. [CrossRef]
23. Raphael, A.; Dubinsky, Z.; Netanyahu, N.S.; Iluz, D. Deep Neural Network Analysis for Environmental Study of Coral Reefs in the Gulf of Eilat (Aqaba). *BDCC* **2021**, *5*, 19. [CrossRef]
24. Wang, H.; Fu, X.; Zhao, C.; Luan, Z.; Li, C. A Deep Learning Model to Recognize and Quantitatively Analyze Cold Seep Substrates and the Dominant Associated Species. *Front. Mar. Sci.* **2021**, *8*, 775433. [CrossRef]
25. Ternon, Q.; Danet, V.; Thiriet, P.; Ysnel, F.; Feunteun, E.; Collin, A. Classification of Underwater Photogrammetry Data for Temperate Benthic Rocky Reef Mapping. *Estuar. Coast. Shelf Sci.* **2022**, *270*, 107833. [CrossRef]
26. Diesing, M.; Green, S.L.; Stephens, D.; Lark, R.M.; Stewart, H.A.; Dove, D. Mapping Seabed Sediments: Comparison of Manual, Geostatistical, Object-Based Image Analysis and Machine Learning Approaches. *Cont. Shelf Res.* **2014**, *84*, 107–119. [CrossRef]
27. Mohamed, H.; Nadaoka, K.; Nakamura, T. Assessment of Machine Learning Algorithms for Automatic Benthic Cover Monitoring and Mapping Using Towed Underwater Video Camera and High-Resolution Satellite Images. *Remote Sens.* **2018**, *10*, 773. [CrossRef]
28. Wicaksono, P.; Aryaguna, P.A.; Lazuardi, W. Benthic Habitat Mapping Model and Cross Validation Using Machine-Learning Classification Algorithms. *Remote Sens.* **2019**, *11*, 1279. [CrossRef]
29. Cui, X.; Liu, H.; Fan, M.; Ai, B.; Ma, D.; Yang, F. Seafloor Habitat Mapping Using Multibeam Bathymetric and Backscatter Intensity Multi-Features SVM Classification Framework. *Appl. Acoust.* **2021**, *174*, 107728. [CrossRef]
30. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
31. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004; ISBN 978-0-521-54051-3.
32. Geisz, J.K.; Wernette, P.A.; Esselman, P.C.; Morris, J.M. *Autonomously Collected Benthic Imagery for Substrate Prediction, Lake Michigan 2020–2021*; U.S. Geological Survey: Reston, VA, USA, 2024. [CrossRef]
33. Fincham, J.I.; Wilson, C.; Barry, J.; Bolam, S.; French, G. Developing the Use of Convolutional Neural Networking in Benthic Habitat Classification and Species Distribution Modelling. *ICES J. Mar. Sci.* **2020**, *77*, 3074–3082. [CrossRef]
34. Gómez-Ríos, A.; Tabik, S.; Luengo, J.; Shihavuddin, A.; Krawczyk, B.; Herrera, F. Towards Highly Accurate Coral Texture Images Classification Using Deep Convolutional Neural Networks and Data Augmentation. *Expert Syst. Appl.* **2019**, *118*, 315–328. [CrossRef]
35. Elith, J. Machine Learning, Random Forests and Boosted Regression Trees. In *Quantitative Analyses in Wildlife Science*; Wildlife Management and Conservation; Johns Hopkins University Press: Baltimore, MD, USA, 2019; p. 281. ISBN 978-1-4214-3107-9.
36. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array Programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef] [PubMed]
37. Bradski, G. The OpenCV Library. *Dr. Dobb's J. Softw. Tools* **2000**, *120*, 122–125.
38. The Pandas Development Team. Pandas-Dev/Pandas: Pandas 2020. Available online: https://zenodo.org/records/10697587 (accessed on 5 September 2023).

39. van der Walt, S.; Schönberger, J.L.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J.D.; Yager, N.; Gouillart, E.; Yu, T. Scikit-image: Image processing in Python. *PeerJ* **2014**, *2*, e453. [CrossRef] [PubMed]
40. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [CrossRef]
41. Haralick, R.M.; Shanmugam, K.; Dinstein, I. Textural Features for Image Classification. *IEEE Trans. Syst. Man Cybern.* 1973; SMC-3, 610–621. [CrossRef]
42. Wang, L.; He, D.-C. Texture Classification Using Texture Spectrum. *Pattern Recognit.* **1990**, *23*, 905–910. [CrossRef]
43. Strang, G. Wavelets. *Am. Sci.* **1994**, *82*, 250–255.
44. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *arXiv* **2015**, arXiv:1603.04467.
45. Keras 2015. Available online: https://keras.io (accessed on 5 September 2023).
46. Zhang, L.; Tanno, R.; Xu, M.-C.; Jin, C.; Jacob, J.; Ciccarelli, O.; Barkhof, F.; Alexander, D.C. Disentangling Human Error from the Ground Truth in Segmentation of Medical Images. In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Virtual, 6–12 December 2020; Neural Information Processing Systems Foundation, Inc. (NeurIPS): Vancouver, BC, Canada, 2021.
47. Ontario Geological Survey 1:250 000 Scale Bedrock Geology of Ontario 2011. Available online: https://www.geologyontario.mndm.gov.on.ca/mndmfiles/pub/data/records/MRD126-REV1.html (accessed on 5 September 2023).
48. Schruben, P.G.; Arndt, R.E.; Bawiec, W.J.; King, P.B.; Beikman, H.M. *Geology of the Conterminous United States at 1:2,500,000 Scale a Digital Representation of the 1974 P.B. King and H.M. Beikman Map*; Data Series; Release 2, 1998; U.S. Geological Survey: Reston, VA, USA, 1998.
49. Shihavuddin, A.S.M.; Gracias, N.; Garcia, R.; Gleason, A.; Gintert, B. Image-Based Coral Reef Classification and Thematic Mapping. *Remote Sens.* **2013**, *5*, 1809–1841. [CrossRef]
50. Stokes, M.D.; Deane, G.B. Automated Processing of Coral Reef Benthic Images: Coral Reef Benthic Imaging. *Limnol. Oceanogr. Methods* **2009**, *7*, 157–168. [CrossRef]
51. Mohammed, R.; Rawashdeh, J.; Abdullah, M. Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. In Proceedings of the 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 7–9 April 2020; pp. 243–248.
52. Ellen Marsden, J.; Casselman, J.M.; Edsall, T.A.; Elliott, R.F.; Fitzsimons, J.D.; Horns, W.H.; Manny, B.A.; McAughey, S.C.; Sly, P.G.; Swanson, B.L. Lake Trout Spawning Habitat in the Great Lakes—A Review of Current Knowledge. *J. Great Lakes Res.* **1995**, *21*, 487–497. [CrossRef]
53. Buscombe, D. Shallow Water Benthic Imaging and Substrate Characterization Using Recreational-Grade Sidescan-Sonar. *Environ. Model. Softw.* **2017**, *89*, 1–18. [CrossRef]
54. Galloway, A.; Brunet, D.; Valipour, R.; McCusker, M.; Biberhofer, J.; Sobol, M.K.; Moussa, M.; Taylor, G.W. Predicting Dreissenid Mussel Abundance in Nearshore Waters Using Underwater Imagery and Deep Learning. *Limnol. Ocean Methods* **2022**, *20*, 233–248. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

geomatics

MDPI

Article

Vector-Algebra Algorithms to Draw the Curve of Alignment, the Great Ellipse, the Normal Section, and the Loxodrome

Thomas H. Meyer

Department of Natural Resources and the Environment, University of Connecticut, 1376 Storrs Rd., U4087, Storrs, CT 06269, USA; thomas.meyer@uconn.edu

Abstract: This paper recasts four geodetic curves—the great ellipse, the normal section, the loxodrome, and the curve of alignment—into a parametric form of vector-algebra formula. These formulas allow these curves to be drawn using simple, efficient, and robust algorithms. The curve of alignment, which seems to be quite obscure, ought not to be. Like the great ellipse and the loxodrome, and unlike the normal section, the curve of alignment from point **A** to point **B** (both on the same ellipsoid) is the same as the curve of alignment from point **B** to point **A**. The algorithm used to draw the curve of alignment is much simpler than any of the others and its shape is quite similar to that of the geodesic, which suggests it would be a practical surrogate when drawing these curves.

Keywords: geometric geodesy; curve of alignment; normal section; geodesic; loxodrome; great ellipse; algorithms

1. Introduction

In a geographic information system (GIS), linear and polygonal features are usually defined by only their vertices, so the programmer is assigned the task of deciding which curve to use to connect the points; that is, the task of defining the implied edge [1]. If the edge is to be a straight line in "the real world", then it could be found conceptually by stretching a chalk line between the endpoints and snapping it down, leaving the trace of the curve on the ground—or, for geomatics, on the reference ellipsoid that is used by the geodetic datum for the mapping. Such a curve is called a geodesic, being "the curve that does not veer to the left nor to the right". Then, the geodesic would be projected onto the map using the forward map-projection formulas.

The formulas for calculating a geodesic curve on an ellipsoid (hereafter just called a geodesic) are very complicated [2–8], and there are many special cases. For example, there are four geodesics connecting any two antipodal points on the Equator: two geodesics following the Equator itself, one to the east and the other to the west, and two others following the northwards and southwards meridians connecting the points. There are other peculiar cases as well, such as the "lift-off" condition [9] for which two points on the Equator are connected by either one or three geodesics depending on how far apart they are. Since there is not always exactly one geodesic connecting two points, and given the programming challenge of implementing them, a programmer might want to consider other curves that are very nearly geodesics but simpler to implement: normal sections, great ellipses, loxodromes, and curves of alignment.

This paper presents a single, unified approach for drawing these curves that is based on a parametric representation using vector algebra. (It would be ideal to provide an algorithm for the geodesic as well; however, there does not appear to be a way to implement general geodesics with this approach.) Parametric solutions have already been found for the great ellipse [10,11], but these approaches use geodetic coordinates for their parameters, which is very helpful for navigation but complicates the formulas if the goal is only to draw the curve. The direct and indirect problems for loxodromes were provided by Meyer

and Rollins [12], but their formulations were also parameterized by geodetic coordinates, resulting in similarly complicated formulas [3] and Deakin [13] report formulas for points along a curve of alignment. Testing revealed that the method below produces the same points as Deakin [13], and the method below is simpler, in the opinion of the author.

Notice that drawing these curves between two points is a separate problem from the typical "direct" and "indirect" problems of geodesy. The former takes a point, an azimuth, and a distance, and the implied forward point is to be determined. The latter takes two points, and the forward and backwards azimuths, and their separation is to be determined. One can use the indirect problem to find the forward azimuth and the separation between the points, and then use the direct problem's solution parametrically as a function of distance to enumerate the points along the curve to be drawn.

The terminology and notation for the remainder of the paper are now given. The lengths of an ellipsoid's semi-major and semi-minor axes are a and b, and they are related by the first eccentricity as $b^2 = \left(1 - \epsilon^2\right) a^2$. The geodetic latitude is $-\pi/2 \leq \phi \leq \pi/2$, longitude is $-\pi \leq \lambda \leq \pi$, and geodetic height is h. See Appendix A for other formulas. The geocentric Cartesian coordinate axes are X, Y, and Z, with Z being the Z-axis of the reference ellipsoid. The X- and Y-axes are perpendicular to Z and to each other, and they also define the Equatorial Plane. The Prime Meridian is the section formed by the X–Z plane. (The reader will recall that a section of an ellipsoid is the curve formed by the intersection of a plane and the ellipsoid.)

The general idea of the algorithms below is to produce a set of XYZ points along the straight-line segment (i.e., the chord) connecting the endpoints of the curve to be drawn. There are simple relationships between these points, which are in the interior of the ellipsoid, and their images on the ellipsoid. The relationship depends upon which curve is desired. An XYZ point, or vector, will be denoted by $\mathbf{P} = \left(X_P, Y_P, Z_p\right)$. A point in geodetic longitude, latitude, and height (LBH) will be denoted by $\mathbf{P} = \left(\lambda_{\overline{P}}, \phi_P, h_P\right)$.

2. Curve of Alignment

The curve of alignment can be visualized in this way (c.f. [13] (Deakin, 2009)). Consider two points of interest, **A** and **B**, on the ellipsoid, and imagine connecting them with a straight line that penetrates through the ellipsoid. Now, imagine another line segment that is perpendicular to the connecting line and that rises outwards, emerging perpendicularly (normal) to the ellipsoid. This rising line is free to slide along the connecting line and it remains normal to the ellipsoid as it moves. The curve of alignment is the locus of points at the intersection of the rising line and the near side of the ellipsoid.

2.1. Properties

The curve of alignment is undefined if **A** and **B** are antipodal.

Since **A** and **B** are on an ellipsoid, their curve of alignment spans less than 180° of longitude.

The curve of alignment from **A** to **B** is the same as the curve of alignment from **B** to **A**.

Any portion less than 180° of the Equator is a curve of alignment and is also a geodesic, as is any such portion of a meridian. A portion of any parallel other than the Equator is not a curve of alignment. (The curve of alignment connecting two points at the same latitude will pass polewards of their common parallel; see Figure 1).

The curve of alignment is not in general a planar curve. In fact, it is the intersection of a hyperbolic paraboloid and an ellipsoid [3], p. 67.

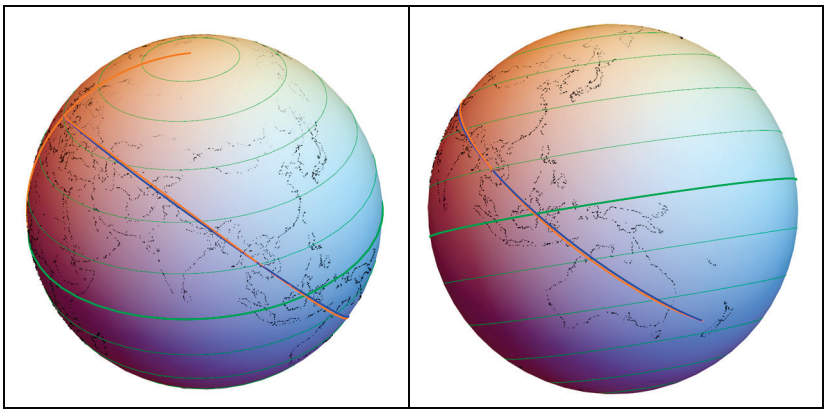


Figure 1. The curve of alignment (orange) and the geodesic (blue) from (165° E, 40° S) and (0° E, 45° N) on the WGS 84 reference ellipsoid. The Equator appears in green and the Prime Meridian in orange.

2.2. Algorithm

The algorithm used to draw curve of alignment is based on geodetic coordinate-system conversions. The algorithm depends on the idea that changing a point's geodetic height only—not geodetic longitude or latitude—moves the point in a direction normal to the ellipsoid. The connecting line segment is given by the difference of **A** and **B** in geocentric Cartesian coordinates, and we formulate this segment parametrically. The rising segment is found by converting a point on the connecting segment from *XYZ* into geodetic longitude, latitude, and height, and then setting the geodetic height to zero. This determines a point on the ellipsoid as the other endpoint of the rising segment such that the rising segment is perpendicular to the ellipsoid. The following six steps make this algorithm explicit.

1. Some ellipsoid is specified, such as a geodetic reference ellipsoid. This provides the length of the semi-major axis a and the value of the (first) eccentricity squared ϵ^2. (See Appendix A for the formulas and parameter values.)
2. Two points of interest, **A** and **B**, not antipodal and on the ellipsoid are chosen. (Note: **A** and **B** are on the ellipsoid so their geodetic heights are zero.) Their coordinates are required in geocentric Cartesian coordinates: $\mathbf{A} = (X_A, Y_A, Z_A)$ and $\mathbf{B} = (X_B, Y_B, Z_B)$. If given in geodetic longitude and latitude, they can be readily converted to geocentric (see Appendix A).
3. Compute the vector **v** = **B**–**A.**
4. The points of the straight-line segment from **A** to **B** are given in parametric form by $\mathbf{p}(t) = \mathbf{A} + t\,\mathbf{v}$ for $0 \leq t \leq 1$.
5. Denote the conversion from geocentric Cartesian coordinates to geodetic longitude, latitude, and height by γ. Then, $\gamma(\mathbf{p}) = (\lambda, \phi, h)$. This conversion depends on a and ϵ^2. Define a variant of γ called $\overline{\gamma}$ such that $\overline{\gamma}(\mathbf{p}) = (\lambda, \phi, 0)$, where, for some h, $\gamma(\mathbf{p}) = (\lambda, \phi, h)$, i.e., $\overline{\gamma}(\mathbf{p})$ returns the same as $\gamma(\mathbf{p})$ but with h set to zero.
6. The curve of alignment is the curve $\overline{\gamma}(\mathbf{p}(t))$ for $0 \leq t \leq 1$. In other words, the curve of alignment is the set of geodetic points given by the conversion of all the points $\mathbf{p}(t)$ from geocentric Cartesian to geodetic, setting their geodetic heights to zero.

An implementation in *Mathematica*™ produced the image in Figure 1, which shows the curve of alignment (red) and the geodesic (green) from (0° E, 45° N) to (165° E, 40° S) on the WGS 84 reference ellipsoid. The arc length of the geodesic is 18,669,335.84 m, and the arc length of the curve of alignment is 18,671,843.56 m, which is 2507.72 m (134 ppm) longer. The arc length of the geodesic was computed using [5], and the arc length of the curve of alignment was computed by recursively bisecting the curve into linear segments and summing the lengths of the segments until the length converged to submillimeter levels. In this example, the curve of alignment starts out north of the geodesic, crosses exactly once near the middle, and is south thereafter.

Notes:

- The vector **v** is equally well defined as $\mathbf{v} = \mathbf{A} - \mathbf{B}$, but then $\mathbf{p}(t) = \mathbf{B} + t\,\mathbf{v}$. The same curve of alignment results either way.
- Pseudocode for the curve of alignment is given below. The algorithm assumes there is a function named XYZ_to_LBH that converts *XYZ* coordinates to geodetic longitude, latitude, and height *LBH*. This code is assumed to be written in a language with built-in vector algebra operators such as *Mathematica* or *Matlab*, or with a vector-algebra library such as *NumPy*.

  A := (XA, YA, ZA) # FROM point in *XYZ*
  B := (XB, YB, ZB) # TO point in *XYZ*
  t: floating point scalar in [0, 1]
  V := **B** − **A** # vector subtraction
  pt := **A** + t * **V** # **pt** will be inside the ellipsoid
  p2 := XYZ_TO_LBH(**pt**) # convert **pt** to LBH $(\lambda_{p2},\ \phi_{p2}, h_{P2})$.
  CoA := $(\lambda_{p2},\ \phi_{p2}, 0)$ # set h_{P2} to zero.

3. Normal Sections

Normal sections can be defined with a single point on an ellipsoid and an azimuth [14]. There is a vector that is normal to the ellipsoid at this point. The sectioning plane contains the point and the normal vector, and the sectioning plane is oriented by the azimuth. The normal section is the intersection of that plane and the ellipsoid. This definition is not well suited for the purposes of this paper, so an alternative based on two points is provided. A normal section from **A** to **B** is a section formed from the intersection of a plane containing **A**, **B**, and another point **V**, where **VA** is normal to the ellipsoid at **A** (see Figure 2). There are many choices for **V**, but it is convenient and helpful to choose **V** to be the point on the *Z*-axis intercepted by the inwards-directed normal vector at **A** (details are given below).

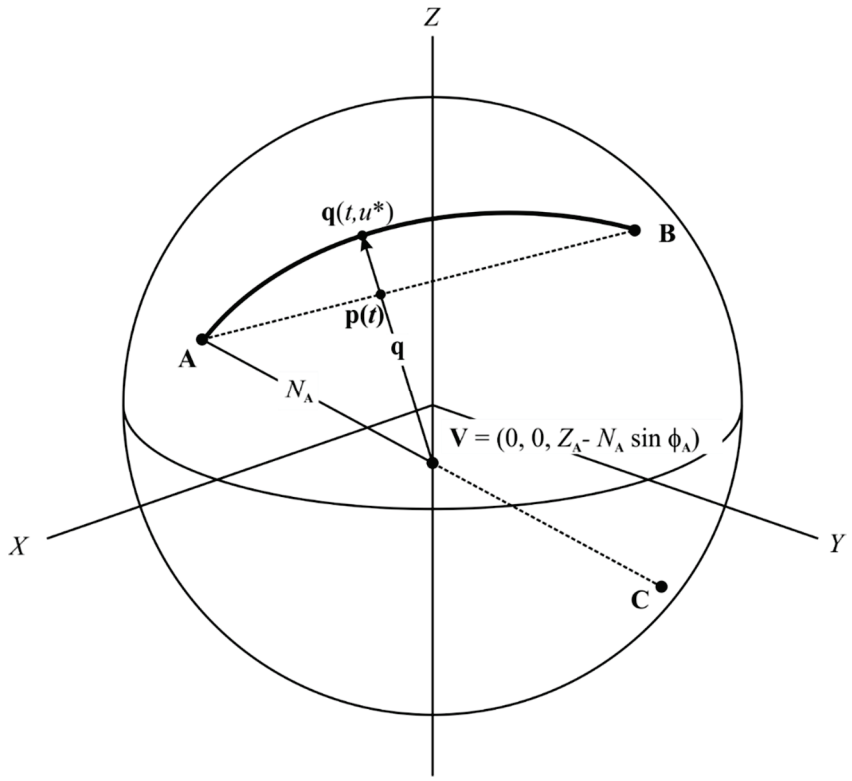


Figure 2. The construction of the (shorter limb of the) normal section from **A** to **B**.

3.1. Properties

The normal section is a planar curve. On an ellipsoid, the normal section is the entire ellipse (a circle for the Equator), but it is common practice to consider the normal section to be the shorter limb from **A** to **B**. The algorithm below produces the shorter limb, which is consistent with the goal of this paper.

A, **B**, and **V** will not define a normal section if they are collinear, but selecting **B** to be antipodal to **A** is not a problem (generally), as will now be shown. Point **C** in Figure 2 is the point where **VA** is extended to intersect the ellipsoid on the hemisphere opposite **A**. **C** is collinear with **A** and **V**, but **C** is not antipodal to **A** unless **A** is on the Equator or is a Pole (antipodal points are on a line through the origin). **VA** is normal to the ellipsoid at **A**, but **VA** does not generally contain the origin due to the ellipsoid having non-zero eccentricity. Therefore, the normal section is uniquely defined if **A** and **B** are antipodal but not on the Equator and not Poles. In that case, there is a singularity at the exact midpoint, which is at the origin and thus does not possess a unique normal vector.

The normal section from **A** to **B** is not the same as the normal section from **B** to **A** unless **A** and **B** are at the same latitude or at the same longitude.

The Equator and the meridians are normal sections. Any parallel other than the Equator is not a normal section (the normal section connecting two distinct points at the same latitude other than zero will pass polewards of their common parallel; see Figure 3).

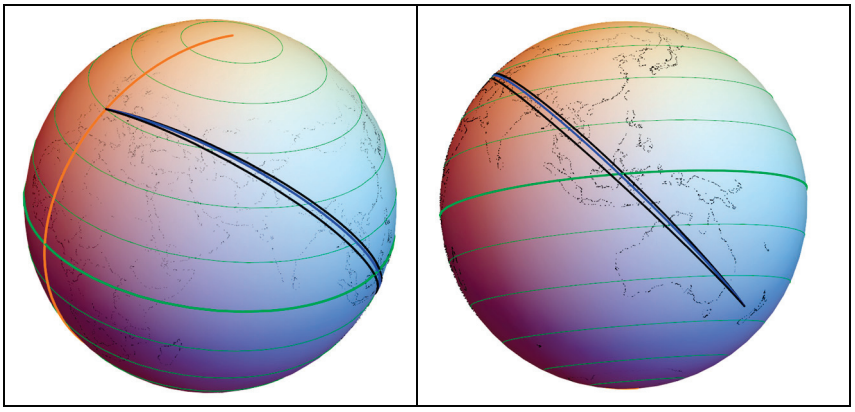


Figure 3. The two normal sections (black) and the geodesic (blue) between (0° E, 45° N) and (165° E, 40° S) on the WGS 84 reference ellipsoid. The Equator appears in green and the Prime Meridian in orange.

The normal section from **A** to **B** does not have the same arc length as the normal section from **B** to **A** unless $\phi_A = \phi_B$ or $\phi_A = -\phi_B$.

3.2. Algorithm

The algorithm used to draw a normal section is similar to that for curve of alignment, but it is a little more complex, so it warrants a figure to help explain it (see Figure 2). Some of the steps given here are expounded upon below in the notes.

1. Some ellipsoid is specified, such as a geodetic reference ellipsoid. This provides the length of the semi-major axis a and the value of the (first) eccentricity squared ϵ^2. (See Appendix A for the formulas and parameter values.)
2. There is a point **V** on the Z-axis at a distance $N_\mathbf{A}$ from **A**, where $N_A = a/\sqrt{1 - \epsilon^2 \sin^2 \phi_A}$ is the radius of curvature in the prime vertical at **A**. **V**'s geocentric coordinates are $(0,\ 0,\ Z_\mathbf{A} - N_\mathbf{A} \sin \phi_\mathbf{A}) = (0,\ 0, -N_\mathbf{A}\ \epsilon^2 \sin \phi_\mathbf{A})$ and **VA** is normal to the ellipsoid at **A**. The plane **AVB** contains **VA** and so **AVB** is the normal sectioning plane at **A**.
3. A point of the straight-line segment (chord) from **A** to **B** is given in parametric form by $\mathbf{p}(t) = \mathbf{A} + t\ (\mathbf{B} - \mathbf{A})$ for $0 \leq t \leq 1$.

4. Any scalar multiple of a vector from **V** to $\mathbf{p}(t)$ is in the normal sectioning plane defined by **AVB**. Let **q** denote a vector in the normal sectioning plane defined by $\mathbf{q}(t,u) = \mathbf{V} + u(\mathbf{p}(t) - \mathbf{V})$, where u is a positive real number.
5. There is one value for u, say u*, such that the geodetic height of $\gamma(\mathbf{q}(t,u^*))$ is zero for $u \geq 0$. The point $\mathbf{q}(t,u^*)$ is on the normal-section curve from **A** to **B** because $\mathbf{q}(t,u^*)$ is in the sectioning plane and the geodetic height of $\gamma(\mathbf{q}(t,u^*))$ is zero. Notice that this is different from the curve of alignment: there, the geodetic height is *set* to zero, but here, a value for u must be found that produces a geodetic height of zero. Setting the height to zero does not work for normal sections because **q** is not generally normal to the ellipsoid. The points on an ellipsoid satisfy $\frac{x^2}{a^2} + \frac{y^2}{a^2} + \frac{z^2}{(1-\epsilon^2)a^2} = 1$, so $\mathbf{q}(t,u^*)$ must also. Define $\boldsymbol{n} = \mathbf{p}(t) - \mathbf{V}$. Substituting in $\mathbf{q}(t,u^*)$'s components and recalling that $V_x = V_y = 0$ gives:

$$\frac{(u^*\, n_x)^2}{a^2} + \frac{(u^*\, n_y)^2}{a^2} + \frac{(V_z + u^*\, n_z)^2}{(1-\epsilon^2)a^2} = 1$$

which is a quadratic in u^*. After some algebra, u^* can be computed from

$$u^* = \frac{-\kappa_1 + \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}}{2\kappa_2} \tag{1}$$

where

$$\kappa_0 = a^{-2}\left(1-\epsilon^2\right)^{-1} V_z^2 - 1$$

$$\kappa_1 = 2a^{-2}\left(1-\epsilon^2\right)^{-1} n_z V_z$$

$$\kappa_2 = a^{-2}\left(n_x^2 + n_y^2 + \left(1-\epsilon^2\right)^{-1} n_z^2\right)$$

6. The normal section from **A** to **B** is the curve $\gamma(\mathbf{q}(t,\ u^*))$ for $0 \leq t \leq 1$. In other words, the normal section from **A** to **B** is the set of points given by the set of all the points $\mathbf{q}(t,u^*)$ converted from geocentric Cartesian to geodetic coordinates.

Notes:

- The normal-section curve is parameterized by t, the same as the curve of alignment.
- In general, each t will have its own u*.
- The point on the normal section is given by $\mathbf{V} + u^*(\mathbf{p}(t) - \mathbf{V})$.
- The vector $\mathbf{q}(0,\ u)$ is normal to the ellipsoid but no other vector $\mathbf{q}(t,u)$ for $0 < t \leq 1$ is also normal unless **A** and **B** have the same latitude, and then $\mathbf{q}(0,u)$ and $\mathbf{q}(1,u)$ are both normal (at **A** and **B**).
- Pseudocode for the normal section is given below, with the same assumptions as for the curve of alignment.

A := (XA, YA, ZA) # starting point in XYZ
B := (XB, YB, ZB) # ending point in XYZ
t: floating point scalar in [0, 1]
$\mathbf{p}_t := \mathbf{A} + t(\mathbf{B} - \mathbf{A})$
$\mathbf{V} := \left(0,\ 0, -N_A\, \epsilon^2 \sin\phi_A\right)$ `# get` ϕ_A `from XYZ_TO_LBH(A)`
$\mathbf{n} := \mathbf{p}_t - \mathbf{V}$
$\kappa_0 := a^{-2}\left(1-\epsilon^2\right)^{-1} V_z^2 - 1$
$\kappa_1 := 2a^{-2}\left(1-\epsilon^2\right)^{-1} n_z V_z$
$\kappa_2 := a^{-2}\left(n_x^2 + n_y^2 + \left(1-\epsilon^2\right)^{-1} n_z^2\right)$ # N.B. $\kappa_2 > 0$
$u^* := -\kappa_1 + \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}\,/\,(2\kappa_2)$
$\mathbf{p_{NS}} := \mathbf{V} + u^*(\mathbf{p}_t - \mathbf{V})$

The two normal sections (black) and the curve of alignment (red) between (0° E, 45° N) and (165° E, 40° S) on the WGS 84 reference ellipsoid are shown in Figure 3. The normal sections have arc lengths of (N–S) 18,669,545.69 m and (S–N) 18,670,163.62 m, making the N–S section 617.94 m longer than the other. The lengths of the normal sections were computed with recursive subdivision, as above.

4. Great Ellipses

For non-antipodal points **A** and **B**, a great ellipse from **A** to **B** is a section formed from the intersection of a plane containing **A**, **B**, and the origin; therefore, the great ellipse is available from the normal-section algorithm simply by setting **V** = (0,0,0). The equation for $\mathbf{q}(t, u^*)$ simplifies to $\mathbf{q}(t, u^*) = u^* \, \mathbf{p}(t)$. The value for:

$$u^* = a^2 \sqrt{(1-\epsilon^2)/((1-\epsilon^2)(n_x^2 + n_y^2) + n_z^2)}$$

The algorithm is otherwise unchanged.

Properties

1. The great ellipse is a planar curve.
2. The great ellipse is the entire ellipse (a circle for the Equator). Every great ellipse contains **A**'s antipodal point by construction; however, the antipodal point cannot be used to define the great ellipse because, then, **A**, **B**, and **V** would be collinear. The algorithm below produces the shorter limb between **A** and **B**, which is consistent with the goal of this paper.
3. The great ellipse from **A** to **B** is the same as the great ellipse from **B** to **A**.
4. The Equator and the meridians are great ellipses. Any parallel other than the Equator is not a great ellipse (the great ellipse connecting two distinct points at the same latitude other than zero will pass poleward of their common parallel; see Figure 4).

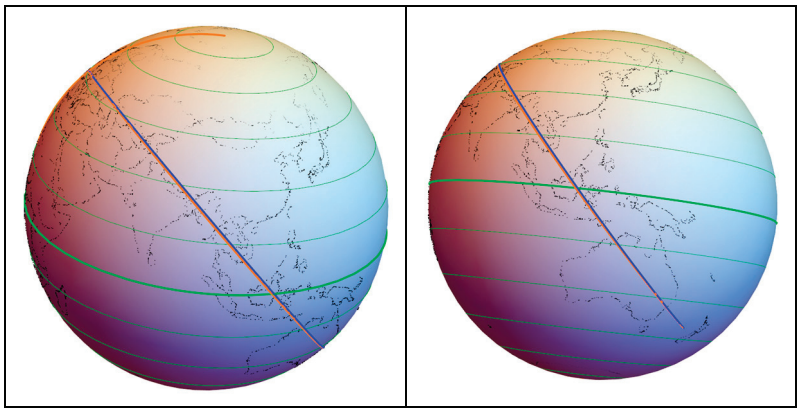


Figure 4. Great ellipse (orange) and the geodesic (blue) between (0° E, 45° N) and (165° E, 40° S) on the WGS 84 reference ellipsoid. The Equator appears in green.

Figure 4 shows the great ellipse, the curve of alignment, and the normal sections between (0° E, 45° N) and (165° E, 40° S) on the WGS 84 reference ellipsoid. The arc length of the great ellipse is 18,669,406.160 m, as computed by recursive subdivision. For this example, the curve of alignment starts out north of the great ellipse, then crosses once near the middle, and remains south thereafter, which is also true for the geodesic.

5. Loxodromes

The loxodrome is a curve of constant azimuth [12], so all the parallels of constant latitude (i.e., "parallels") are loxodromes. A loxodrome is drawn as a straight line on a

Mercator-projection map, which permits a method of drawing loxodromes in geodetic coordinates using this paper's approach.

Properties

1. The parallels and meridians are loxodromes.
2. A loxodrome is a spiral if its starting azimuth is not a cardinal direction. In that case, loxodromes spiral infinitely many times around both Poles but do not reach them—yet, they have finite length. For example, for the WGS 84 ellipsoid and azimuth 45°, the total length works out to be 28,289,831.17 m [12].
3. The loxodrome from **A** to **B** is the same as the loxodrome from **B** to **A**.
4. The loxodrome is not a planar curve in general.
5. The loxodrome is defined if **A** and **B** are antipodal.

The algorithm used to draw loxodromes given below is based on the loxodromes being straight lines when drawn using a Mercator projection. See [15] or [3] for traditional implementations of the Mercator projection. However, Rollins and Meyer [16] provide a more succinct implementation, which is used here. Set the central meridian of the projection λ_0 to be the average of λ_A and λ_b, $\overline{\lambda}$, which avoids issues that arise if the loxodrome crosses 180°. However, the arithmetic average of two angles can fail, so let $s = \sin(\lambda_A) + \sin(\lambda_B)$ and $c = \cos(\lambda_A) + \cos(\lambda_B)$. Then, $\lambda_0 = \overline{\lambda} = \text{atan2}(c, s)$, and

$$x = a(\lambda - \lambda_0)$$

$$y = aq$$

$$q = \text{atanh}(\sin\,\phi) - \epsilon\text{atanh}(\epsilon\sin\phi)$$

where atan2 is a 2-argument arctangent function, atanh is the inverse hyperbolic tangent, x is the Mercator easting, y is the Mercator northing, and q is isometric latitude. The Mercator forward mapping formulas are applied to **A** and **B**, producing Mercator grid coordinates (x, y), and the vector **v** is given by their difference, as before. Points $\mathbf{p}(t)$ in the Mercator grid are computed as a multiple of **v** and then converted back to geodetic coordinates. The inverse formulas are from [16], as provided in Appendix A. The following six steps make this algorithm explicit.

1. Some ellipsoid is specified, such as a geodetic reference ellipsoid. This provides the length of the semi-major axis a and the value of the (first) eccentricity squared ϵ^2. (See Appendix A for the formulas and parameter values.)
2. Two points of interest on the ellipsoid are chosen. Their coordinates are required in geodetic coordinates; e.g., $\mathbf{A} = (\lambda_\mathbf{A}, \phi_\mathbf{A})$ and $\mathbf{B} = (\lambda_\mathbf{B}, \phi_\mathbf{B})$. If given in geocentric Cartesian coordinates, they can be readily converted to geodetic.
3. Using the forward mapping equations (see above), convert **A** and **B** into Mercator coordinates $\mathbf{a} = (x_a, y_a)$ and $\mathbf{b} = (x_b, y_b)$.
4. Compute the vector $\mathbf{v} = \mathbf{b} - \mathbf{a}$.
5. The points of the straight-line segment from **a** to **b** are given in parametric form by $\mathbf{p}(t) = \mathbf{a} + t\,\mathbf{v}$ for $0 \leq t \leq 1$.
6. Denote the inverse Mercator mapping equations (see Appendix A) by γ^M. Then, $\gamma^M(\mathbf{p}(t)) = (\lambda, \phi)$.
7. The loxodrome is curve $\gamma^M(\mathbf{p}(t))$ for $0 \leq t \leq 1$. In other words, the loxodrome is the set of points given by the conversion of all the points $\mathbf{p}(t)$ from Mercator grid coordinates to geodetic coordinates.

An implementation in *Mathematica*™ produced Figure 5, which shows the geodesic (green) and loxodrome (blue) from (0° E, 45° N) to (165° E, 40° S) on the WGS 84 reference ellipsoid. The arc length of the geodesic is 18,669,335.840 m, and the arc length of the loxodrome is 19,066,164.69 m, as computed by recursive subdivision, which is 396,828.850 m longer.

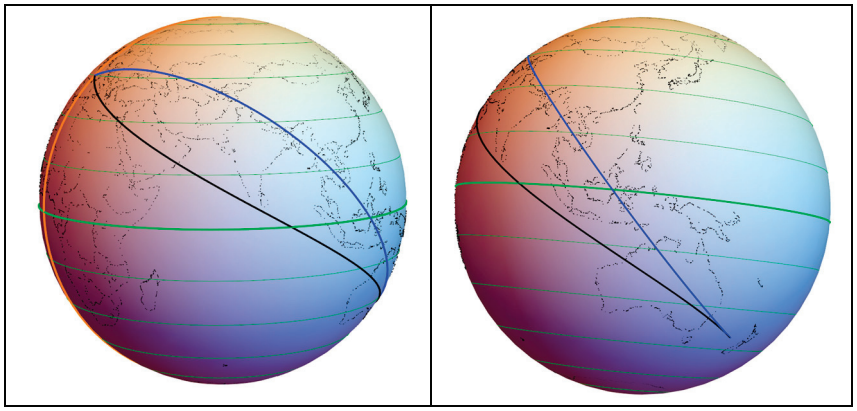


Figure 5. The geodesic (blue) and loxodrome (black) from (0° E, 45° N) to (165° E, 40° S) on the WGS 84 reference ellipsoid. The Equator appears in green and the Prime Meridian in orange.

Notes:

- The vector **v** is equally well defined as $\mathbf{v} = \mathbf{a} - \mathbf{b}$, but then $\mathbf{p}(t) = \mathbf{b} + t\,\mathbf{v}$. The same loxodrome results either way.
- The inverse formula for latitude requires finding a fixed point. This can be cast as a root-finding problem by finding the latitude such that the difference of the input latitude minus the output latitude equals zero.
- Pseudocode for the loxodrome is given below. The reference-ellipsoid object RE is assumed to have a method ecc1 that returns the first eccentricity of the reference ellipsoid; otherwise, the same assumptions as above pertain.

Define a function $Mf(\lambda, \phi) \to (x, y) := (a\lambda_A, a(\text{atanh}(\sin\phi_A) - \epsilon\text{atanh}(\epsilon\sin\phi_A)))$.
Define a function $Mr(x, y) \to (\lambda, \phi)$ to be the reverse Mercator mapping formulas. See Appendix A.

t: floating point scalar in [0, 1]
$\mathbf{A} := (\boldsymbol{\lambda}_\mathbf{A}, \boldsymbol{\phi}_\mathbf{A})$ # starting point
$\mathbf{a} := \boldsymbol{Mf}(\mathbf{A})$ # forward Mercator mapping formulas
$\mathbf{B} := (\boldsymbol{\lambda}_\mathbf{B}, \boldsymbol{\phi}_\mathbf{B})$ # ending point
$\mathbf{b} := \boldsymbol{Mf}(\mathbf{B})$ # forward Mercator mapping formulas
$\mathbf{v} := \mathbf{b} - \mathbf{a}$ # vector subtraction
$\mathbf{p}_t := \mathbf{a} + t\mathbf{v}$ # linear interpolation
$\mathbf{p}_L := Mr(\mathbf{p}_t)$ # reverse Mercator mapping formula

6. Summary

Four new algorithms are presented that draw loxodromes, great ellipses, curves of alignment, and normal sections. The algorithms are parameterized so that a simple implementation based on vector algebra is available, given a supporting geometric geodesy library to calculate the coordinate system. There are many, more or less equivalent, choices for the conversion from geocentric Cartesian coordinates to geodetic longitude, latitude, and height [17,18]. However, for long lines, the chord between the beginning and ending points can come near to the geocenter, which can cause some algorithms to fail. For such situations, a robust algorithm such as [19] is recommended. Also, the algorithms herein are language neutral and do not depend on external packages. However, it should be noted that efficient algorithms and packages exist for geodesics [20] when the actual geodesic is required.

The examples herein were computed using the WGS84 reference ellipsoid. Ref. [21] showed that the minute difference in the flattening of these ellipsoids cannot result in a difference in coordinates at the millimeter level. So, these examples are properly illustrative for GRS80, as well.

Funding: This research received no external funding.

Data Availability Statement: Two Python 3.9 scripts that implement the algorithms (curves.py) and provide some testing and usage demonstrations (curves_demo.py) are available at https://github.com/thomas6meyer/geodetic_curves.git. Last accessed 4 March 2024.

Acknowledgments: The author is grateful to the anonymous reviewers who helped improve this manuscript.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A

For any geodetic reference ellipsoid, the semi-major axis a is a defining parameter. The ellipsoid's shape is given either by the semi-minor axis b, by the flattening f, or by the square of the first eccentricity ϵ^2. See [22] for a list of the parameter values defining geodetic reference ellipsoids. The value for a is 6,378,137 m for the GRS 80 and WGS 84 reference ellipsoids. The values for f are 1/298.257222101 and 1/298.257223563 for the GRS 80 and WGS 84 reference ellipsoids. (This minute difference in the flattenings creates no practical difference in the positions [21].

The flattening of an oblate ellipse is:

$$f = (a - b)/b$$

The first eccentricity squared of an oblate ellipsoid is:

$$\epsilon^2 = (a^2 - b^2)/a^2 = 2f - f^2$$

The semi-minor axis b can be computed from either the flattening or the first eccentricity.

The radius of curvature in the prime vertical is $N = a/\sqrt{1 - \epsilon^2 \sin^2 \phi}$.

Conversion from geodetic to geocentric:

$$X = (N + h)\cos\lambda\cos\phi$$

$$Y = (N + h)\sin\lambda\cos\phi$$

$$Z = ((1 - \epsilon^2)N + h)\sin\phi$$

but for $h = 0$:

$$Z = (1 - \epsilon^2)\, N \sin\phi$$

The Mercator forward (direct) mapping equations take geodetic coordinates (λ, ϕ) to Mercator grid coordinates (x, y). They are as follows:

$$x = a\,(\lambda - \lambda_0)$$

$$y = a\,q$$

$$q = \operatorname{atanh}(\sin\phi) - \epsilon \operatorname{atanh}(\epsilon \sin\phi)$$

There is no exact algebraic expression that provides the geodetic latitude directly from the Mercator northing, so either iterations or approximations are needed. The formulas below follow from the mathematics in [16] and they use iteration. For longitude,

$$\lambda = \lambda_0 + x/a$$

For latitude,

$$q = y/a$$

$$s = \text{Limit of } s^{(0)},\ s^{(1)},\ s^{(2)},\ \ldots$$

$$\phi = \text{asin}(s)$$

where:

$$s^{(0)} = \tanh(q)$$

$$s^{(i+1)} = \tanh(q + \epsilon \text{atanh}(\epsilon\ s^{(i)})$$

and asin(...) is the inverse sine function. The iteration is avoidable by using truncated infinite series: "first calculate the conformal latitude chi based on the inverse of the spherical Mercator, then use the method found in https://arxiv.org/pdf/2212.05818.pdf to calculate geodetic latitude" (accessed 17 April 2024) (anonymous reviewer 2024).

References

1. Kallay, M. Defining edges on a round Earth. In Proceedings of the 15th International Symposium on Advances in Geographic Information Systems, ACM GIS 2007, Seattle, DC, USA, 7–9 November 2007. [CrossRef]
2. Vincenty, T. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Surv. Rev.* **1975**, *23*, 88–93. [CrossRef]
3. Thomas, P.D. *Conformal Projections in Geodesy and Cartography, Special Publication 251*; U.S. Naval Oceanographic Office: Washington, DC, USA, 1979.
4. Sjöberg, L.E. New solutions to the direct and indirect geodetic problems on the ellipsoid. *Z. Für Geodasie Geoinf. Und Land Manag.* **2006**, *131*, 35–39.
5. Rollins, C.M. An integral for geodesic length. *Surv. Rev.* **2010**, *42*, 20–26. [CrossRef]
6. Karney, C.F. Algorithms for geodesics. *J. Geod.* **2013**, *87*, 43–55. [CrossRef]
7. Panou, G.; Korakitis, R. Geodesic equations and their numerical solution in Cartesian coordinates on a triaxial ellipsoid. *J. Geod. Sci.* **2019**, *9*, 1–12. [CrossRef]
8. Karney, C.F. Geodesics on an arbitrary ellipsoid of revolution. *J. Geod.* **2024**, *98*, 4. [CrossRef]
9. Thien, G. A Solution to the Inverse Problem for Nearly Antipodal Points on the Equator of the Ellipsoid of Revolution. Master's Thesis, The Ohio State University, Columbus, OH, USA, 1967.
10. Earle, M.A. A vector solution for navigation on a great ellipse. *J. Navig.* **2000**, *53*, 473–481. [CrossRef]
11. Tseng, W.-K.; Guo, J.-L.; Liu, C.-P.; Wu, C.-T. The vector solutions for the great ellipse on the spheroid. *J. Geod.* **2012**, *6*, 103–109. [CrossRef]
12. Meyer, T.H.; Rollins, C.M. The Direct and Indirect Problem for Loxodromes. *Navigation* **2011**, *58*, 1–6. [CrossRef]
13. Deakin, R.E. *The Curve of Alignment on an Ellipsoid*; RMIT University, School of Mathematical & Geospatial Sciences: Melbourne, Australia, 2009.
14. Deakin, R.E. *The Normal Section Curve on an Ellipsoid*; RMIT University, School of Mathematical & Geospatial Sciences: Melbourne, Australia, 2009.
15. Snyder, J.P. *Map Projections—A Working Manual*; Professional Paper 1395p; U.S. Geological Survey: Washington, DC, USA, 1987; 1395p.
16. Rollins, C.M.; Meyer, T.H. Conformal mapping at elevation with implications for low-distortion projections. *Surv. Rev.* 2024, *in press*.
17. Claessens, S.J. Efficient transformation from Cartesian to geodetic coordinates. *Comput. Geosci.* **2019**, *133*, 104307. [CrossRef]
18. Blewitt, G. An improved equation of latitude and a global system of graticule distance coordinates. *J. Geod.* **2024**, *98*, 6. [CrossRef] [PubMed]
19. Panou, G. Cartesian to geodetic coordinates conversion on an oblate spheroid using the bisection method. *Comput. Geosci.* **2019**, *133*, 104308. [CrossRef]
20. Karney, C.F. GeographicLib. 2015. Available online: http://geographiclib.sourceforge.net (accessed on 17 April 2024).
21. Rollins, C.M.; Meyer, T.H. The effect of delta epsilon on XYZ coordinates. *Surv. Rev.* **2012**, *44*, 282–284. [CrossRef]
22. Meyer, T.H. *Introduction to Geometrical and Physical Geodesy: Foundations of Geomatics*; ESRI Press: Redlands, CA, USA, 2010; 246p.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

Improved A* Navigation Path-Planning Algorithm Based on Hexagonal Grid

Zehua An, Xiaoping Rui * and Chaojie Gao

School of Earth Sciences and Engineering, Hohai University, Nanjing 210098, China; anzehua@hhu.edu.cn (Z.A.); 221309020004@hhu.edu.cn (C.G.)
* Correspondence: ruixp@hhu.edu.cn; Tel.: +86-136-7110-6220

Abstract: Navigation systems are extensively used in everyday life, but the conventional A* algorithm has several limitations in path planning applications within these systems, such as low degrees of freedom in path planning, inadequate consideration of the effects of special regions, and excessive nodes and turns. Addressing these limitations, an enhanced A* algorithm was proposed using regular hexagonal grid mapping. First, the approach to map modeling using hexagonal grids was described. Subsequently, the A* algorithm was refined by optimizing the calculation of movement costs, thus allowing the algorithm to integrate environmental data more effectively and flexibly adjust node costs while ensuring path optimality. A quantitative method was also introduced to assess map complexity and adaptive heuristics that decrease the number of traversed nodes and increase the search speed. Moreover, a turning penalty measure was implemented to minimize unnecessary turns on the planned paths. Simulation results confirmed that the improved A* algorithm exhibits superior performance, which can dynamically adjust movement costs, enhance search efficiency, reduce turns, improve overall path planning quality, and solve critical path planning issues in navigation systems, greatly aiding the development and design of these systems and making them better suited to meet modern navigation requirements.

Keywords: navigation system; path planning; grid map; A* algorithm; regular hexagonal

1. Introduction

Navigation systems are crucial tools for travel, and path planning is a fundamental aspect of their functionality. The primary objective of path planning is to identify a feasible and optimized route from the starting point to the endpoint [1]. In practice, complex terrain and traffic conditions often lead to disparities between planned paths and actual road conditions. Consequently, the development of advanced path-planning methods has emerged as a prominent topic in academic research.

Existing path-planning methods are primarily composed of two main components: (1) the construction of abstract maps and (2) the path-search algorithm [2]. Map construction encompasses various techniques, including graphical representation [3], navigation mesh representation [4,5], and the grid method [6,7]. Currently, most navigation systems use graphical representations that rely on vector-based road networks for path planning. However, this approach involves pre-set traversable paths with limited degrees of freedom. Notably, certain passable areas, such as grasslands, squares, and mountains, were excluded from the planning scope despite applying to various scenarios.

In contrast, the navigation mesh abstracts the original map into a collection of triangular meshes, utilizing the connection relationships between the triangular meshes for path planning [8]. Although this method provides a more accurate representation of a map, its structure is complex and challenging to implement and modify, and the construction process consumes a considerable amount of memory. Consequently, it is primarily used in 3D scenes, such as video games [9].

The grid method divides a scene evenly into a series of grids by considering each grid as a node in an abstract graph. Subsequently, the search algorithm navigates through the abstract graph [10]. This approach provides significant flexibility, covers the entire traffic area effectively, and addresses all aspects within the realm of path planning. Its principle is straightforward, and its implementation is relatively simple.

In a two-dimensional plane, squares, equilateral triangles, and regular hexagons uniformly cover all areas [11]. Conventional grid-based methods rely on square grids. Figure 1 illustrates the movement pattern in different grid maps, where the blue nodes indicate the current node, the white nodes indicate the neighboring nodes and the black arrows indicate the directions of movement. As illustrated in Figure 1, in a square grid, the target can move from the current node to a neighboring node in either four or eight directions. If a diagonal movement is not allowed, there are only four directions: forward, backward, left, and right, each with a movement cost of one unit. The 8-directional square grid map includes four diagonal directions, resulting in two different movement costs: 1 unit and $\sqrt{2}$ units. Therefore, when calculating the paths based on this map, it is necessary to determine the category of the forward direction, which makes the calculation extremely complex. However, the hexagonal grids feature only six target directions, each with a movement cost of one unit.

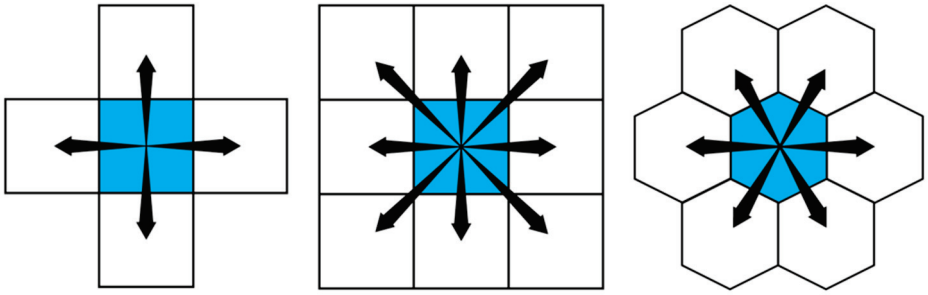


Figure 1. Movement pattern in different grid maps.

Figure 2 illustrates the path on different grid maps, where blue indicates the path nodes, black indicates the obstacle nodes and the black arrows indicate the routes. As depicted in Figure 2, the path exhibits a zigzag pattern with lower smoothness in a 4-neighbor square grid map. In an 8-neighbor square grid map, although the path is smoother, diagonal crossings may cause the map to approach or even contact obstacles, potentially leading to collisions in practical scenarios. However, hexagonal grid maps do not have these limitations. The paths on the hexagonal grid maps demonstrated smoother trajectories devoid of any rigid sensations. Compared to the 8-neighbor square map, the planned path in the hexagonal grid maintained a greater distance from the obstacles. Additionally, each node possesses a single adjacency type with neighboring nodes, which simplifies the algorithm's implementation and enhances search efficiency. Therefore, this study investigates the adoption of hexagonal grid maps over conventional square-grid maps to enhance the A* algorithm.

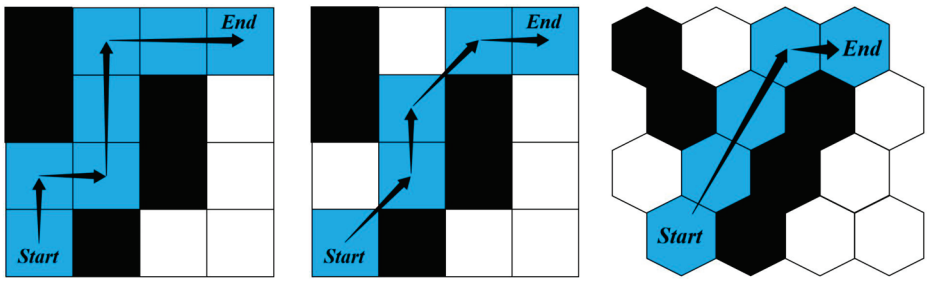


Figure 2. Path on different grid maps.

Various algorithms have been applied in the field of pathfinding, including breadth-first search (BFS) [12], depth-first search (DFS) [13], the Dijkstra algorithm [14], genetic algorithms [15], ant colony algorithms [16], and the A* algorithm [17]. Among the algorithms described above, the A* algorithm is a heuristic pathfinding method that merges the best features of Dijkstra's and greedy algorithms. Renowned for its effectiveness in solving optimal path problems in static environments [18], as early as 1984, Pearl discussed the completeness and optimality of the A* algorithm in his book *Intelligent Search Strategies for Computer Problem Solving*. However, issues such as a fixed heuristic function and the limited impact of heuristics often limit the adaptability of the A* algorithm to the influence of special areas on the movement costs during path searches. These issues also result in the traversal of many unnecessary nodes, accompanied by the creation of numerous turning points.

Since the introduction of the A* algorithm in 1968, scholars worldwide have continuously researched and optimized it. Current improvements to the A* algorithm primarily focus on optimizing distance calculation methods, enhancing heuristic functions, and secondary path optimization. The optimization of distance calculation methods [19,20] involves refining the estimation methods for the distance between the current point and the destination. By fully and rationally using the destination's location information, this approach reduces the path length and enhances the efficiency of the algorithm. However, this method prioritizes distance cost as the main criterion and overlooks the impact of special terrain on robot mobility. Enhancements to the heuristic function [19–24] involve weighting the components of the heuristic or incorporating new guiding metrics that address issues such as slow search speeds and excessive node traversals. However, these improvements lack adaptability to different maps and often result in suboptimal solutions in complex environments. Secondary path optimization [25] involves the additional processing of paths planned by the A* algorithm, which shortens the path length and improves smoothness. However, this approach requires the design of additional path-smoothing methods, thereby increasing the complexity of the algorithm and reducing the pathfinding efficiency. Moreover, the aforementioned improvements did not consider the impact of special terrain on the algorithm but rather focused more on obstacles.

In practical applications, maps often encompass unique terrains, rendering the aforementioned algorithmic enhancements inadequate for complex environments. To address this issue, the current study emphasizes the use of regular hexagonal grid maps and integrates terrain information to guide navigation pathfinding. This integration aims to enhance algorithmic efficiency and improve the smoothness of the planned paths. Thus, conventional A* algorithms, constrained by their inability to accommodate special terrain, excessive node traversals, and numerous inflection points on raster maps, were adapted. An improved A* algorithm was proposed, featuring an enhanced heuristic function with newly designed elements and the incorporation of environmental information. This refinement not only preserves the optimality of the planned path but also enhances efficiency, environmental adaptability, and path quality.

2. Conventional A* Algorithm

Pathfinding algorithms can be classified into blind and heuristic categories based on whether they use heuristics. Blind algorithms conduct searches based on predetermined strategies without considering the characteristics of the map or destination. Examples include the breadth-first search (BFS) and Dijkstra's algorithm. However, heuristic algorithms utilize estimated costs to reach the destination, calculating heuristic values to expedite the search. Representative examples include greedy and A* algorithms. The most typical one is the greedy algorithm, which selects the node with the minimum estimated cost to reach the destination at each expansion step. The A* algorithm combines the features of both Dijkstra's and Greedy's algorithms with its heuristic function, defined as

$$f(n) = g(n) + h(n) \tag{1}$$

where n represents the node currently traversed during the search process; $g(n)$ represents the cost from the starting point to the current node; $h(n)$ represents the estimated cost from the current node to the destination, which serves as the heuristic value in the A* algorithm; and $f(n)$ is the heuristic function of the A* algorithm.

$$g(n) = \sum_{i=0}^{n} S(i) \tag{2}$$

$S(i)$ represents the distance movement for the corresponding node. For example, $S(n)$ represents the actual distance of a single step from the parent node $(n-1)$ to the current node n. Therefore, $g(n)$ is the sum of the distances for each segment of the planned path. In a hexagonal grid where the distance between adjacent nodes is uniform, you can set $S(i) = 1$ to facilitate the operation of the algorithm.

Heuristic algorithms require the calculation of distances between nodes, and the distance calculation involves the assistance of a coordinate system. Common hexagonal grid coordinate systems include cubic, offset, and axial coordinates. Among these, the cubic coordinate system is easy to understand, and its distance calculation formula is simple and clear. Therefore, a cubic coordinate system was adopted for the distance calculation. In contrast to the x- and y-axes in the Cartesian coordinates for square grid maps (as shown in Figure 3), the cubic coordinate system has three axes: q, r, and s, representing the grid coordinates as (q, r, s).

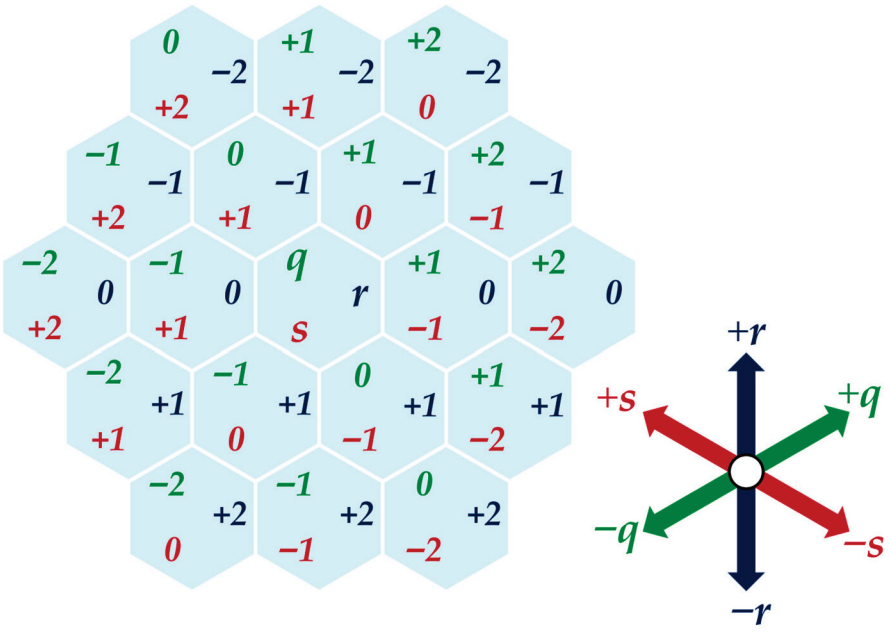


Figure 3. Diagram of cubic coordinate system.

Common distance calculation methods include the Euclidean, Manhattan, and diagonal distances, among which the Manhattan distance, also known as a taxicab or city block distance, measures the total absolute difference along each axis between two points in a grid-based environment. In hexagonal grids, it is intuitive to measure the distance between two nodes by using the number of steps per unit. Therefore, the Manhattan distance was used to estimate the movement costs and calculate the distance from the current node to the destination.

For grid $\boldsymbol{A}$ with coordinates (q_1, r_1, s_1) and grid $\boldsymbol{B}$ with coordinates (q_2, r_2, s_2), the Manhattan distance between $\boldsymbol{A}$ and $\boldsymbol{B}$ is expressed as:

$$D(\boldsymbol{A}, \boldsymbol{B}) = \frac{|\boldsymbol{A}.q - \boldsymbol{B}.q| + |\boldsymbol{A}.r - \boldsymbol{B}.r| + |\boldsymbol{A}.s - \boldsymbol{B}.s|}{2} \tag{3}$$

During the search process of the A* algorithm, each time it finds the node with the smallest heuristic function value for expansion, the specific steps of the algorithm are as follows:

(1) Create *OpenList* and *ClosedList* collections; add the starting point to the *OpenList*, and set the starting point's $g(n) = 0$.
(2) Find the node with the smallest $f(n)$ value in the *OpenList*. If the *OpenList* is empty, the search fails, and the search ends.
(3) Add the node to the *ClosedList* and traverse all neighbor nodes of the current node that are not in the *ClosedList*.
    a If the neighbor node is not on *OpenList*, let the neighbor node's movement cost $g(n + 1)$ equal $g(n) + 1$, and set the current node as the parent node of this neighbor node.
    b If the neighbor node is in *OpenList* and the calculated $g(n)$ from the current node to this neighbor node is smaller than the previously calculated $g(n)$ from the parent node, then update the value of $g(n)$, and the current node is set as the parent node of this neighbor node; if it is greater, then it is not updated.
(4) Repeat Steps 2 and 3 until the node found in Step 2 is the endpoint.

The parent node was traced sequentially from the endpoint. If the parent node is the starting point, it ends, and the path planning result is obtained.

A flowchart of the A* algorithm is shown in Figure 4.

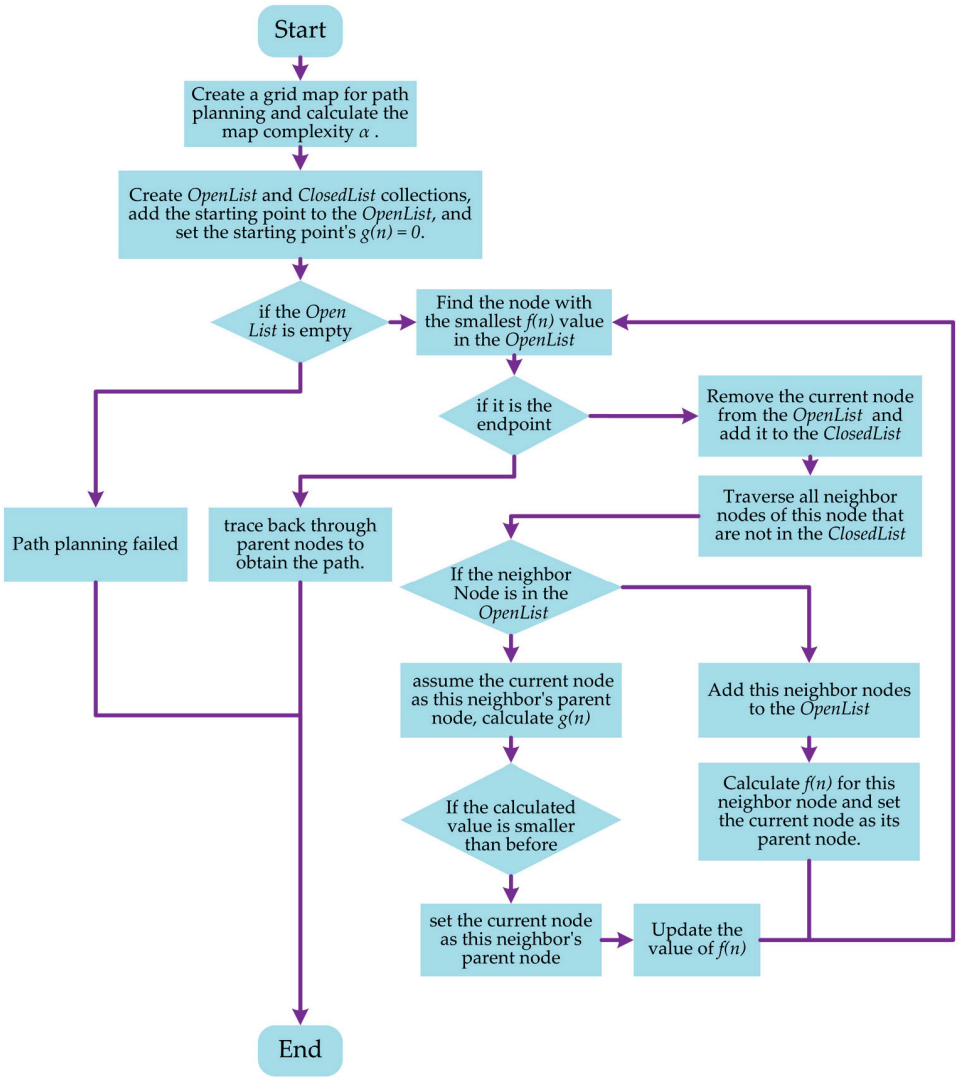


Figure 4. Flowchart of the A* algorithm.

3. Improved A* Algorithm

The conventional A* algorithm is not influenced by special terrain. During the path search process, problems such as a high number of traversed nodes, numerous turning points, and a large total turning angle occur. The algorithm is enhanced by implementing a flexible cost strategy, an adaptive heuristic function, turning the penalty function, and incorporating the impact of special terrain into the path search. While ensuring path optimality, this approach reduces the number of traversed nodes, enhances algorithm efficiency, and enables quick and stable planning of inflection points and total turning angles, resulting in a more reasonable navigation path.

3.1. Flexible Setting of Costs

The conventional grid method typically defines two attributes: passability and inaccessibility. In practical applications, path planning maps contain some special areas, such as grasslands and hills, which are often included. These areas are traversable; however, because of their specific characteristics, distance cannot be used directly as the cost of movement in these areas. Therefore, we define the grid nodes in three states: (1) Ordinary nodes, (2) nodes in special areas, and (3) obstacle nodes. The node states are stored when abstracting maps. As the algorithm traverses a new node, if the node is a regular node, the movement cost for that node is determined by the distance, specifically set to one. If the node is located within a special area, the transit cost is determined by considering both the distance and the effects of the special area. If a node is an obstacle node, it is not expanded. Further improvements include enhancing the calculation method for $g(n)$ and introducing the influencing factor K into the heuristic function. K represents the difficulty of passing through a special area. If node i is situated in a region influenced by factor K, then the single-step movement cost $S'(i)$ from the node i's parent node to node i is defined as follows:

$$S'(i) = 1 + K \quad (K > -1) \tag{4}$$

The constant term 1 represents the basic distance cost of moving to node i; the overall movement cost, or the difficulty of passage, is directly proportional to K. When $K = 0$, the node is considered ordinary, and when $K = 1$, the cost of moving to node i is twice that of moving to an ordinary node. When the K value is sufficiently large, node i can be considered located in an impassable obstacle.

Moreover, a path node may be located at the intersection of multiple areas, and the impact of these areas on the cost calculation of the node must be comprehensively considered. If m special areas are superimposed at the location of node i, the expression for $S(i)$ is

$$S'(i) = 1 + \sum_{j=1}^{m} K_j \tag{5}$$

where K_j represents the influencing factor for the special area j, and this formula is used to calculate the cost $g'(n)$ of moving to the current node after being affected.

3.2. Adaptive Heuristics

By incorporating the influence of the parent node $(n - 1)$, the heuristic value from the parent node is added to the current node's heuristic function, increasing the overall proportion of the heuristic value within the heuristic function. This enhancement can accelerate search speed, traverse fewer nodes during the search, and reduce computational load. However, excessively high search speeds may lead to the emergence of locally optimal solutions. To enable the algorithm to accelerate the search speed on simpler maps and adopt a more cautious approach for complex maps, it is necessary to adjust the heuristic weight based on map information, allowing it to adaptively control the search speed. If the number of nodes in the current map is x, the number of obstacle nodes is y, there are m types of special areas, and the number of nodes with corresponding area properties is ε,

with corresponding special area influence factors represented by K, all these parameters determined by map information, then the improved heuristic expression $h'(n)$ is as follows:

$$h'(n) = (1 - \alpha) \times [h(n) + h(n - 1)] \tag{6}$$

$$\alpha = \frac{x}{y + \sum_{i=0}^{m} \frac{\varepsilon_i K_i}{1 + K_i}} \tag{7}$$

In the above equation, $1 - \alpha$ represents the adaptive weight multiplier for the heuristic, and α denotes the map complexity, with a value range of [0, 1]. When the extent of obstacles and special areas is large, the complexity of the map increases, reducing the heuristic value and consequently slowing the search speed.

To verify the impact of the adaptive heuristic values on the search speed, an experiment was conducted on a 30 × 30 hexagonal grid map, and the results are displayed in Figures 2 and 5. Figure 5a shows the search results using the conventional A* algorithm; Figure 5b shows the search results when only the heuristic value from the parent node is introduced; and Figure 5c presents the search results after introducing the adaptive heuristic proposed in this study. In the figures, gray nodes represent ordinary nodes, black nodes represent obstacle nodes, light blue nodes denote path nodes, and orange nodes indicate nodes traversed during the search. The black triangle marks the starting point of the path, and the black pentagon marks the endpoint of the path. And the dark blue line is the planned path. The calculated value of α is 0.276. A comparison of the search results is presented in Table 1.

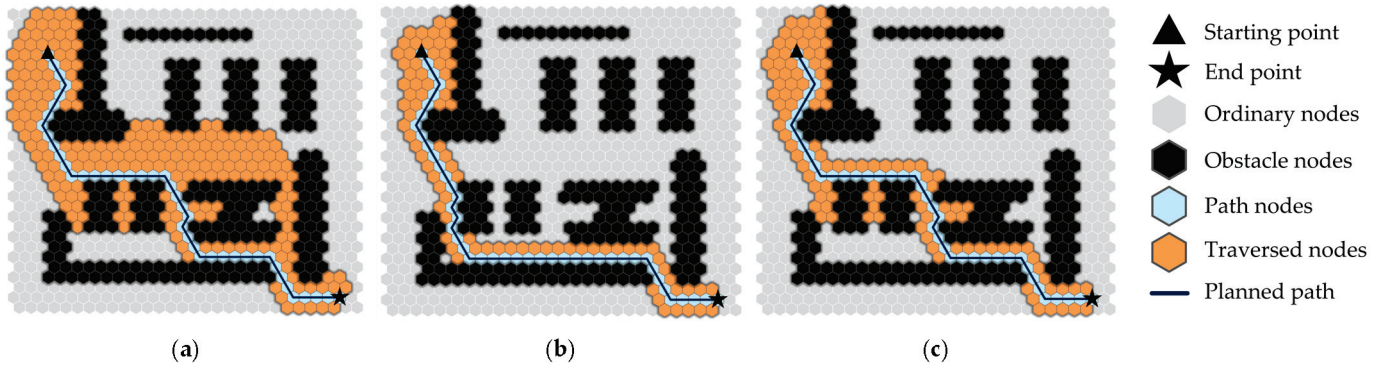


Figure 5. Adaptive heuristic test results. (**a**) Conventional A* algorithm. (**b**) Introducing the influence of the parent node. (**c**) Introducing adaptive heuristics.

Table 1. Comparison of search results.

Test Items	Conventional A* Algorithm	Introducing the Influence of Parent Node	Introducing Adaptive Heuristics
Number of path nodes/number	43	44	43
Number of traversed nodes/node	261	125	149
Number of turning points/piece	9	9	9
Total turning angle/°	540	540	540

The results indicate that, after introducing the heuristic value from the parent node, there is a significant reduction in the number of nodes traversed. As the time required to traverse each node remains the same, the search speed is also improved. However, the planned path loses optimality. However, after incorporating adaptive heuristic values into the heuristic function, the algorithm maintains path optimality, and the number of traversed nodes is significantly reduced, thereby demonstrating the feasibility of this

approach. However, there is still redundancy in the turning points across the various algorithms, necessitating further optimization and improvement.

3.3. Turn Penalty

The formation of redundant turning was due to the low stability of the algorithm during the search process. When faced with multiple paths having the same movement cost, the algorithm fails to select a route with a smaller number of turning points and a smaller total turning angle. Therefore, this study proposes the incorporation of a penalty term to suppress the formation of turning points, thus enabling the algorithm to maintain the direction of path progression and avoid turning.

The direction of the path progression is shown in Figure 6, where the blue node represents the current node, the orange node represents the parent node of the current node, and the gray nodes are the neighboring nodes of the current node, which will be traversed in the next iteration. The black arrow indicates the direction of progression from the parent node to the current node, represented by vector $\boldsymbol{a}$; the red arrow indicates the possible direction of progression from the current node to a new node, represented by vector $\boldsymbol{b}$.

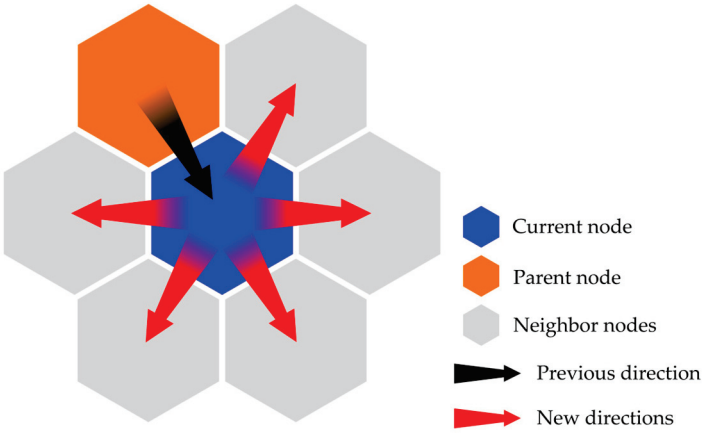


Figure 6. Schematic of path progression direction.

When traversing a new node, there are six directions for expansion, which, compared to the previously planned route, yields four outcomes: 1. Continue along the previous route; 2. The new node direction differed from the previous path direction by 60°; and 3. The new node direction differed from the previous path direction by 120°. It can be observed that the turning angle of the route is related to the angle θ between the previous path direction and the new node direction. Assume that the current node is n, its parent node is $(n-1)$, and the new node is $(n+1)$, then the angle θ is calculated as:

$$cos\ \theta = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{|\boldsymbol{a}| \cdot |\boldsymbol{b}|} \tag{8}$$

where $\boldsymbol{a}$ is the vector from the current node to the new node, and $\boldsymbol{b}$ is the vector from the parent node to the current node.

Add a penalty value of $punish(n+1)$ to each new node; $punish(n+1) = \beta(1 - cos\,\theta)$; the value range of the penalty value is [0, 2a]. The factor β serves as a penalty intensity coefficient, with the penalty intensity being directly proportional to the value of β. Accordingly, the algorithm prioritizes the expansion of new nodes, where the direction of progression remains unchanged, to minimize the number of turns during the search and reduce the number of turning points.

3.4. Final Improved Algorithm

To enable the conventional A* algorithm to adapt to the influence of complex maps, prioritize planning paths in areas with low traffic difficulty, enhance search speed, reduce the number of traversed nodes, and minimize turning points and angles, this study proposes a flexible setting of movement costs, adaptive heuristics, and a turning penalty strategy, ultimately resulting in an improved A* algorithm. First, a regular hexagonal grid map is used for abstraction; the special terrain influence factor information is stored in the map, and α is calculated. The expression for the improved A* algorithm's heuristic function $f'(n)$ is as follows:

$$f'(n) = g'(n) + (1 - \alpha)\,[h(n) + h(n-1)] + punish(n) \tag{9}$$

4. Algorithm Simulation

To verify the performance of the improved A* algorithm proposed in this study, we simulated it using the Unreal Engine 5 (UE5), version 5.0.3. UE5 is a powerful game development engine increasingly used by researchers for developing navigation systems, making it an excellent software tool for the experiments conducted in this study. We constructed an experimental map of size 30 × 30, which included 191 obstacle nodes and 248 grassland nodes, with the influence factor K set to 1. Through calculations using Equation (7), the complexity α of this map was determined to be 0.35. The map and selection of the start and end points are shown in Figure 7. In the figure, gray nodes represent ordinary nodes, light green nodes represent grass nodes, and black nodes represent obstacle nodes. The black triangle marks the starting point of the path, and the black pentagon marks the endpoint of the path.

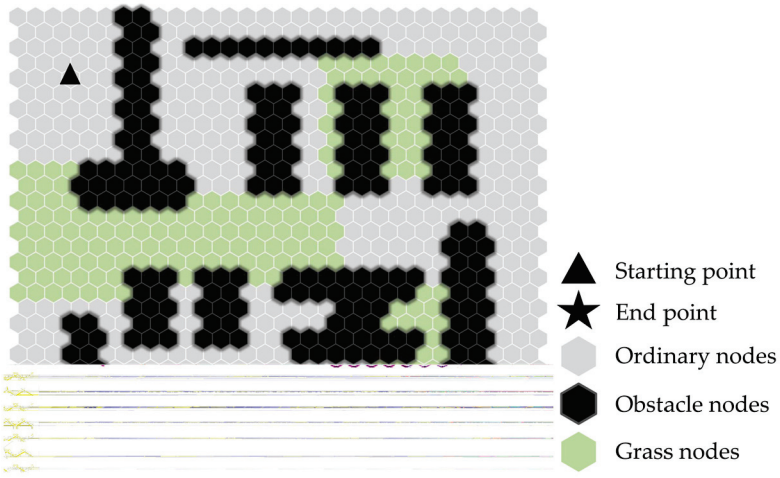


Figure 7. Experimental map.

First, we conducted experiments on the magnitude of the penalty intensity coefficient β under the premise of introducing flexible cost settings and adaptive heuristics. The search results for the algorithms with different β values are listed in Table 2.

Table 2. Performance comparison of different β value algorithms.

β Value	0	0.25	0.5	0.75	1	1.5	2
Number of path nodes/number	44	44	44	44	44	44	44
Path movement cost/step	60	60	60	60	60	60	60
Number of traversed nodes/node	263	261	261	267	267	276	280
Number of turning points/piece	17	12	10	10	10	12	12
Total turning angle/°	1020	720	600	600	600	720	720

From the analysis of the experimental results, the introduction of a turning penalty reduces both the number of turning points and the total turning angle. When β exceeds 0.5, the number of nodes traversed increases because of the excessive penalty, causing the algorithm to traverse more nodes where the direction of progression has not changed rather than nodes closer to the destination. When β is between 0.5 and 1, the path exhibits the fewest turning points and the smallest total turning angle, with a 41% reduction in the number of turning points and a 54% reduction in the number of redundant turning points, demonstrating significant optimization effects. Therefore, we select a value of 0.5 for β in the improved algorithm described in this study.

To evaluate the effectiveness of the proposed algorithm, we conducted simulations for comparison with the conventional A* algorithm, an algorithm featuring flexible cost settings, an algorithm with an adaptive heuristic, and the algorithms described in [19,20]. These simulations were performed on the map designed in the previous section. The resultant paths generated by each algorithm are depicted in Figure 8, where gray indicates ordinary nodes, light green denotes grassland nodes, light blue highlights the paths identified by the algorithms, orange marks ordinary nodes traversed during the search, and green signifies grassland nodes traversed. A comparative analysis of these algorithms is presented in Table 3, which lists their performance metrics.

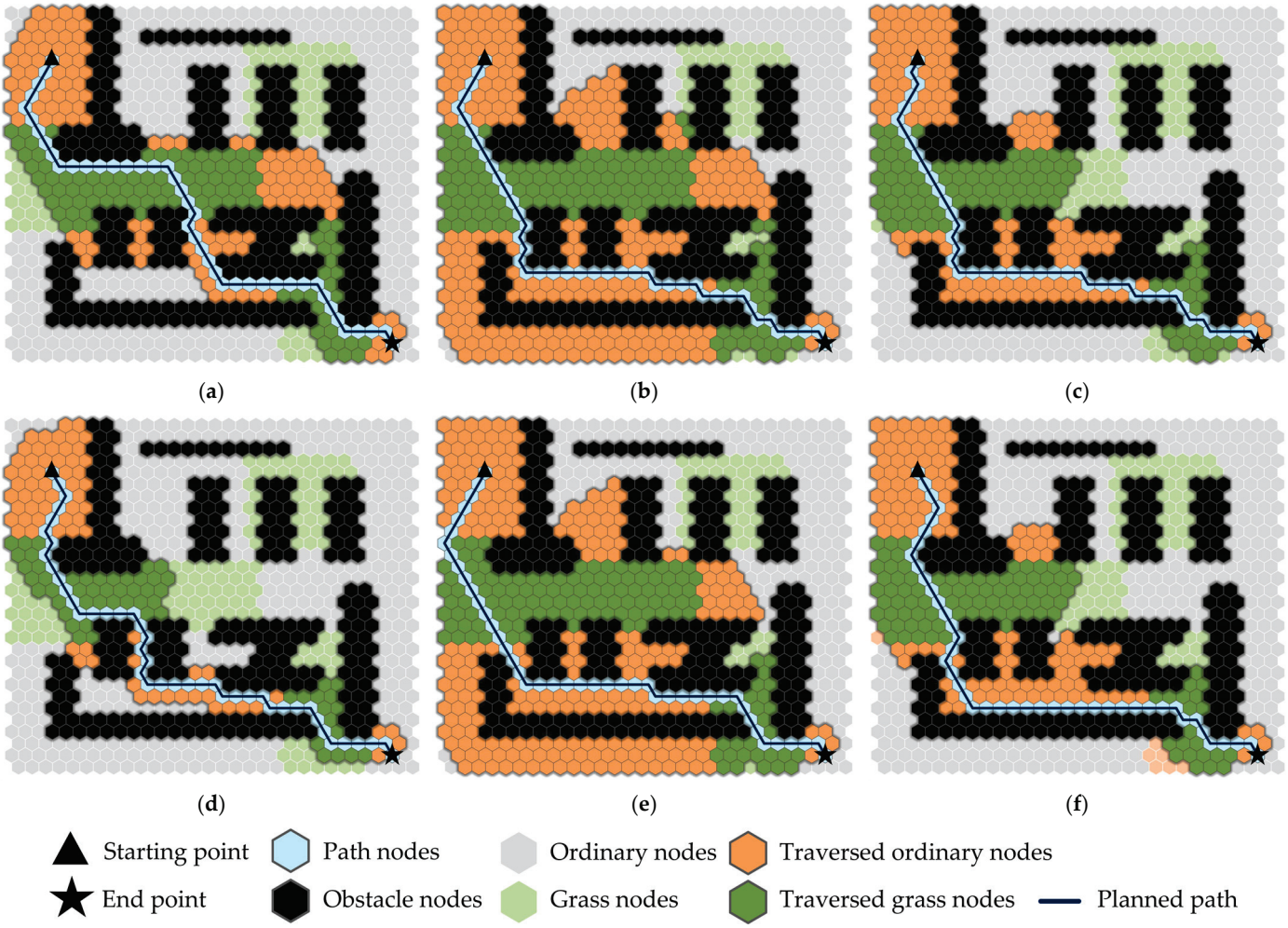


Figure 8. Results from different algorithms. (**a**) Conventional A* algorithm. (**b**) Improved algorithm for flexible cost setting. (**c**) Introduce adaptive heuristics. (**d**) The algorithm mentioned in reference [19]. (**e**) The algorithm mentioned in reference [20]. (**f**) Improved A* algorithm of this study.

Table 3. Performance comparison of different algorithms.

Index	Conventional A* Algorithm	Flexible Setting of Costs A* Algorithm	Introduce Adaptive Amount of Inspiration	The Algorithm Mentioned in Reference [19]	The Algorithm Mentioned in Reference [20]	Improved A* Algorithm of This Study
Number of path nodes/step	43	44	44	44	44	44
Path movement cost/step	66	60	60	63	60	60
Number of traversed nodes/number	261	420	263	172	409	261
Number of turning points/piece	9	15	17	18	7	10
Total turning angle/°	540	900	1020	1080	420	600

It can be observed that the conventional A* algorithm only determines the shortest distance path between the start and end points, traversing long distances through grassy areas, which increases the overall movement cost and loses optimality. After adopting a flexible cost-setting method, the algorithm identifies the path with the lowest cost and achieves optimality; however, owing to the influence of special terrains, the number of traversed nodes and turning points significantly increases. The introduction of adaptive heuristic values maintains path optimality while substantially reducing the number of traversed nodes. However, the A* algorithm already exhibits redundancy in the number of turning points and angles, and the introduction of adaptive heuristic values adds three more turning points, necessitating further optimization.

The algorithm mentioned in [19] uses the average of the Manhattan and Euclidean distances for distance calculation. Although this visually accelerates the approach to the destination, it also results in excessive turning points. Additionally, by incorporating parent node information into the heuristic calculation, a significant reduction in the number of traversed nodes was observed, which was 59.0% lower than that of the A* algorithm with flexible cost settings, significantly enhancing the search speed. However, this was achieved at the expense of sacrificing the optimality and stability of the algorithm. From Figure 8d, it is evident that the algorithm chooses to move across grassy areas in the early part of the path, increasing the movement cost by three and adding three additional turning points, thereby losing path optimality.

The algorithm in [20] introduced the orientation information of the target node into the heuristic function. It first calculates the cosine of the angle between the direction of advance and the target direction, using this cosine value to either increase or decrease the cost based on the presence of obstacles in the forward direction, thus quickly circumventing them. The algorithm in reference [18] is shown to find the optimal path, reducing the number of turning points and the total turning angle by more than half, although the decrease in the number of traversed nodes is minimal, resulting in lower search efficiency.

The improved A* algorithm proposed in this study determines the optimal path on maps with special regions, showing a substantial improvement over the flexible cost-setting A* algorithm in terms of the number of traversed nodes, turning points, and turning angles. Compared to the algorithm in [19], although it traverses more nodes, it plans an optimal path with fewer turning points. Relative to the algorithm in [20], it has more turning points, but the number of nodes traversed differs significantly, demonstrating a significant improvement in the pathfinding efficiency. The analysis in Table 3 shows that the improved A* algorithm outperforms the conventional A* algorithm. Specifically, the improved A* algorithm reduces the overall movement cost by six, a decrease of 9.1% while maintaining path optimality, and reduces the average number of nodes traversed by 21.8%, with an average decrease in the number of turning points and the total turning angle of 25.0%.

To validate the performance of the improved algorithm for different obstacles and grassland areas, we conducted a simulation using a 30×30 map. On this map, obstacles

and grass nodes are generated at random locations based on different probabilities. The conventional A*, flexible cost setting, and improved algorithms presented in this study were simulated under the same starting conditions. The results are summarized in Table 4. Experiments where the obstacle generation probability was set to 0.2 and 0.4, with a grass generation probability of 0, did not include results for the flexible cost-setting algorithm. This is because the flexible cost-setting algorithm is identical to the A* algorithm when no grass is present. Therefore, those results were omitted. Conventional A* algorithm plans shorter paths but ignores the impact of grassy areas, resulting in higher movement costs. By contrast, the improved algorithm maintained the minimum movement cost even when facing changes in obstacles and grassy areas.

Table 4. Comparison of experimental results and data under different maps.

Obstacle Generation Probability		**0.2**	**0.4**	**0**	**0**	**0.2**	**0.2**	**0.4**	**0.4**
Grass Generation Probability		**0**	**0**	**0.2**	**0.4**	**0.2**	**0.4**	**0.2**	**0.4**
Map Complexity α		**0.211**	**0.401**	**0.089**	**0.202**	**0.283**	**0.376**	**0.438**	**0.531**
Conventional A* algorithm	Number of path nodes/step	45	54	41	40	44	40	48	46
	Path movement cost/step	44	53	47	59	48	57	59	60
	Number of traversed nodes/number	191	368	204	198	175	127	166	114
	Number of turning points/piece	18	27	8	8	12	12	24	20
Flexible setting of costs A* algorithm	Number of path nodes/step	—	—	41	41	44	40	50	47
	Path movement cost/step	—	—	41	44	46	47	57	56
	Number of traversed nodes/number	—	—	204	533	165	173	343	182
	Number of turning points/piece	—	—	14	15	14	20	22	26
Improved A* algorithm of this paper	Number of path nodes/step	45	54	41	41	44	40	50	47
	Path movement cost/step	44	53	41	44	46	47	57	56
	Number of traversed nodes/number	106	183	128	220	114	110	172	198
	Number of turning points/piece	6	23	4	15	8	16	22	26

When the number of obstacles and grass nodes was low, the proposed algorithm discussed in this paper traversed fewer nodes and planned paths with significantly fewer turning points than the flexible cost-setting algorithm. This is because the map was not particularly complex at this stage, with paths less obstructed by obstacles and grassy areas. The adaptive heuristic and turning penalty strategies enabled the algorithm to rapidly approach the endpoint, effectively enhancing search efficiency. As the probability of generating obstacles and grass increased, making the map more complex, the difference in the number of nodes traversed and the turning points between the improved and flexible cost-setting algorithms decreased. When the obstacle and grass probabilities reached 0.4, the map became sufficiently complex (complexity exceeded 0.5), making the algorithm more cautious than the conventional A* algorithm during path searches. It traverses more nodes than the flexible cost-setting algorithm to ensure that it determines the path with the lowest movement cost. When the map complexity is high, the space for path selection decreases, making turning points unavoidable. However, the paths planned by the improved algorithm consistently had fewer or equal turning points than those planned by the flexible cost-setting algorithm, enabling smoother path planning.

These experimental analyses demonstrated that the improved A* algorithm can adapt to complex mapping environments. It exhibited good adaptability and stability when faced with maps containing different numbers of obstacles and special terrain. It can plan optimal paths with higher search efficiency and fewer turning points and angles, making it more suitable for path planning in navigation systems.

5. Conclusions

This study focused on path planning within navigation systems and conducted a comparative analysis between square and regular hexagonal grid maps. This study proposed

an improved A* algorithm tailored to hexagonal grids. The following conclusions were drawn from the experimental comparative analysis:

- To address the issues of conventional A* algorithms on grid maps that do not respond to special terrain, feature a large number of traversed nodes, and have many turning points, the movement cost calculation method was improved. By introducing an influencing factor, the movement costs were set flexibly, allowing the path search to fully consider the effects of the special terrain. Subsequently, adaptive heuristics were introduced to adjust the heuristics affected by adding parent nodes using map information, which accelerated the search speed while maintaining optimal path conditions. Finally, a turning penalty was designed to reduce the number of turns in the path.
- The improved algorithm exhibited good adaptability when facing maps with varying numbers of obstacles and special terrains. Under the premise of minimizing path movement costs, it can adjust the search speed based on the map conditions, reducing the number of nodes traversed, turning points, and turning angles, thereby enhancing the algorithm efficiency and path quality and demonstrating remarkable stability.
- The improved A* algorithm proposed in this study optimizes path planning on complex maps with special areas such as grasslands and hills, outperforming conventional A* algorithms and other improved algorithms from the literature. Overall, the path movement costs of the improved algorithm presented in this study decreased by 9.1%. While maintaining path optimality, the average number of traversed nodes was reduced by 21.8%, and the number of turning points and the total turning angle both decreased by an average of 25.0%.

In summary, the improved A* algorithm demonstrated significant superiority and adaptability, providing an effective solution for path planning in navigation systems and producing paths that are more realistic and consistent with common travel practices.

The algorithm primarily targets static path planning with complete environmental information and cannot incorporate dynamically appearing or disappearing obstacles and special terrain into path planning. Moreover, pathfinding cannot be performed when the environmental information is not fully known. Our future research will focus on dynamic and partial path planning to ensure accurate and efficient pathfinding in dynamic environments and situations with limited environmental information.

Author Contributions: Conceptualization, Zehua An and Xiaoping Rui; methodology, Zehua An and Xiaoping Rui; software, Zehua An; validation, Zehua An and Chaojie Gao; formal analysis, Zehua An and Chaojie Gao; investigation, Zehua An; resources, Zehua An and Chaojie Gao; data curation, Zehua An; writing—original draft preparation, Zehua An; writing—review and editing, Zehua An; visualization, Xiaoping Rui and Chaojie Gao; supervision, Xiaoping Rui; project administration, Zehua An; funding acquisition, Xiaoping Rui. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 42376180; No. 41771478).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tomás, L.; Wesley, M. An Algorithm for Planning Collision Free Paths Among Polyhedral Obstacles. *Commun. ACM* **1979**, *22*, 560–570. [CrossRef]
2. Gao, T.; Liu, W. Summarization of intelligent pathfinding-oriented 3D scene. *Comput. Eng. Appl.* **2017**, *53*, 16–22.
3. Teo, T.; Cho, K. BIM-oriented indoor network model for indoor and outdoor combined route planning. *Adv. Eng. Inform.* **2016**, *30*, 268–282. [CrossRef]
4. Lou, X.; Sun, M.; Yang, S. A fine-grained navigation network construction method for urban environments. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *113*, 102994. [CrossRef]

5. Zhu, J.; She, P.; Li, W.; Cao, Y.; Qi, H.; Wang, B.; Wang, Y. Dynamic Planning Method for Indoor-Fire Escape Path Based on Navigation Grid. *J. Southwest Jiaotong Univ.* **2020**, *55*, 1103–1110.
6. Liu, Z.; Zhang, L.; Wei, N.; Kuang, X. Research on gridding and path planning of environmental map. *Ship Sci. Technol.* **2021**, *43*, 141–145.
7. Wang, W.; Ai, T.; Yan, X.; Lu, W. Indoor Route Planning Under Regular Hexagonal Grid Considering. *Geomat. Inf. Sci. Wuhan Univ.* **2020**, *45*, 111–118.
8. Meysami, A.; Cuillière, J.-C.; François, V.; Kelouwani, S. Investigating the Impact of Triangle and Quadrangle Mesh Representations on AGV Path Planning for Various Indoor Environments: With or without Inflation. *Robotics* **2022**, *11*, 50. [CrossRef]
9. Xiao, C.; Hao, S. An Overview of Pathfinding in Navigation Mesh. *Int. J. Comput. Sci. Netw. Secur.* **2012**, *12*, 48–51.
10. Zhu, Q.; Zhang, Y. An Ant Colony Algorithm Based on Grid Methodfor Mobile Robot Path Planning. *Robot* **2005**, *27*, 132–136.
11. Liu, Y.; Zhou, X.; Yan, Y. Research on image processing algorithms on hexagonal grid. *Comput. Eng. Des.* **2001**, *5*, 71–75.
12. Tripathy, H.; Mishra, S.; Thakkar, H.; Rai, D. A Collision-Aware Mobile Robot Navigation in Grid Environment using Improved Breadth First Search. *Comput. Electr. Eng.* **2021**, *94*, 107327. [CrossRef]
13. Gang, T. R-DFS: A Coverage Path Planning Approach Based on Region Optimal Decomposition. *Remote Sens.* **2021**, *13*, 1525. [CrossRef]
14. Wang, J.; Wei, G.; Dong, X. A dynamic fire escape path planning method with BIM. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 10253–10265. [CrossRef]
15. Tu, Y.; Zhao, Y.; Liu, L.; Nie, L. Travel route planning of core scenic spots based on best-worst method and genetic algorithm: A case study. *Manag. Syst. Eng.* **2022**, *1*, 4. [CrossRef]
16. Baker, L. Path Planning of Mobile Robot based on Improved Ant Colony Algorithm. *Sci. Technol. Eng.* **2022**, *22*, 12484–12490. [CrossRef]
17. Hart, P.; Nilsson, N.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
18. Su, L.; Jiang, H. Building Internal Path Planning Algorithm and its Application Reviewed Research. *Geomat. Spat. Inf. Technol.* **2014**, *10*, 105–109.
19. Chen, C. Research on Robot Shortest Path Planning with Improved A* Algorithm. *Comput. Digit. Eng.* **2023**, *51*, 1697–1701.
20. Wu, S.; Huang, Y.; Chen, T. Static Route Planning of Surface Ships Based on Improved A* Algorithm. *Comput. Eng. Appl.* **2022**, *58*, 307–315.
21. Zhang, Y.; Tang, G.; Chen, L. Improved A* Algorithm for Time—Dependent Vehicle Routing Problem. *Control. Eng. China* **2012**, *19*, 750–752, 756.
22. Li, X.; Hu, X.; Wang, Z.; Du, Z. Path Planning Based on Combination of Improved A-STAR Algorithm and DWA Algorithm. In Proceedings of the 2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM), Manchester, UK, 15–17 October 2020; pp. 99–103. [CrossRef]
23. Ju, C.; Luo, Q.; Yan, X. Path Planning Using an Improved A-star Algorithm. In Proceedings of the 2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan), Jinan, China, 23–25 October 2020; pp. 23–26.
24. Zhang, H.; Li, M.; Yang, L. Safe Path Planning of Mobile Robot Based on Improved A* Algorithm in Complex Terrains. *Algorithms* **2018**, *11*, 44. [CrossRef]
25. Zhou, J.; Yang, L.; Zhang, C. Indoor robot path planning based on improved A* algorithm. *Mod. Electron. Tech.* **2022**, *45*, 181–186.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

Integration of Spatial and Co-Existence Relationships to Improve Administrative Region Target Detection in Map Images

Kaixuan Du [1,2], Fu Ren [1,3,*], Yong Wang [2], Xianghong Che [2], Jiping Liu [2], Jiaxin Hou [2] and Zewei You [1]

[1] School of Resource and Environmental Science, Wuhan University, Wuhan 430079, China; dukaixuan@whu.edu.cn (K.D.); you_zw@whu.edu.cn (Z.Y.)
[2] Research Center of Geospatial Big Data Application, Chinese Academy of Surveying and Mapping, Beijing 100830, China; wangyong@casm.ac.cn (Y.W.); chexh@casm.ac.cn (X.C.); liujp@casm.ac.cn (J.L.); 11220844@stu.lzjtu.edu.cn (J.H.)
[3] Key Laboratory of GIS, Ministry of Education, Wuhan University, Wuhan 430079, China
* Correspondence: renfu@whu.edu.cn

Abstract: Administrative regions are fundamental geographic elements on maps, thus making their detection in map images crucial to enhancing intelligent map interpretation. However, existing methods in this field primarily depend on the texture features within the images and do not account for the influence of spatial and co-existence relationships among different targets. In this study, taking the administrative regions of the Chinese Mainland, Taiwan, Tibet, and Henan as test targets, we employed the spatial and co-existence relationships of pairs of targets to improve target detection performance. Firstly, these four regions were detected using a simple Single-Target Cascading detection model based on RetinaNet. Subsequently, the detection results were adjusted with the spatial and co-existence relationships of each pair of targets. The adjusted outcomes demonstrate a significant increase in target detection accuracy, as well as precision (from 0.62 to 0.96) and *F1* score (from 0.76 to 0.88), for the Chinese Mainland target. This study contributes to the advancement of intelligent map interpretation.

Keywords: spatial relationship; co-existence relationship; RetinaNet model; map administrative region; target detection

1. Introduction

Maps, important geographic information carriers representing one of the three common international languages, play a significant role in the expression of geographic space [1] and are applied in various fields of research, including economic activities and trade, industrial production, environmental protection, disaster prevention and mitigation, and epidemic prevention [2–10]. Currently, a large number of raster maps are available on the internet because of advancements in computer science and internet technology; they are derived from scanned historical maps, contemporary atlases, vector-to-raster maps with original vector data missing, hand-drawn maps, etc. The geographic information locked in raster maps is crucial for researchers to understand long-term development in an area without manual on-site data acquisition [11–13].

Map interpretation is important for the development of geospatial intelligence [12]. Administrative regions represent fundamental geographic information for depicting locations where current research is taking place [4] and providing a framework for public administration [9]. The recognition of administrative regions plays a fundamental role in map interpretation. As administrative regions are objects in map images, the research complexities faced in administrative region detection can be transformed into object/target detection in computer vision. However, generally, within administrative region symbols, there are combined cartographic symbols, including points, lines and areas, which makes

the detection of administrative regions in maps more complicated than that of an individual symbol. Most traditional target detection methods first develop hand-crafted features derived from Sobel Canny operator recognition [14,15], Haar-like feature detectors [16], HOG detectors [17], DPMs [18], etc., and then detect the target with shallow classifiers such as SVMs [19] or boosting regressors [20]. Among them, the highest recorded accuracy is only 33.7% (mAP), achieved with DPM-v5 on the VOC07 dataset [21].

With the development of artificial intelligence, deep learning networks have become powerful techniques for image classification [22,23], object detection [24] and object semantic segmentation [25]. In terms of raster map content recognition, most deep learning-based methods have been developed for map classification [26–29], map style classification [30], specific region target detection [31–33] and map textual annotation identification [34,35]. In general, there are two main paradigms in image object detection based on deep learning that have been widely used: two-stage detectors (i.e., R-CNN [36], SPP Net [37], Fast R-CNN [38] and Faster R-CNN [39]) and one-stage detectors (i.e., YOLO [40], SSD [41] and RetinaNet [42]). The former includes two steps, i.e., proposing regions and then classifying and regressing bounding boxes, while the latter directly predicts the bounding boxes and class labels in a single pass [24]. Compared with traditional methods that rely on manually designed features, object detection based on deep learning leverages its learning capability to automatically learn more discriminative feature representations, thereby enhancing the accuracy and robustness of object detection. However, map administrative region detection is rarely performed by using deep learning-based object detection [31–33]. In one study [32], the authors proposed a method based on Faster R-CNN [39] to predict specific region objects with multi-scale feature fusion by using fixed-aspect-ratio anchors. In another study [33], the authors introduced and compared two typical map-based geographic object detection methods based on YOLO and RetinaNet [42] and proved that the latter was superior to the former in accuracy and speed. In a previous study [43], a framework based on CNNs and GCNs for the extraction and recognition of geological map symbols was proposed. In another study [44], a three-stage framework based on deep learning to automatically recognize symbols in geological maps was proposed. Finally, in the study described in [45], the authors compared Faster R-CNN and RetinaNet in the detection of ship objects in maps.

In the region target detection studies discussed above, only the color and texture features of regions were extracted, while the spatial and co-existence relationships among regions were not accounted for. However, different targets in maps have very strong spatial correlations. For example, on maps, the Beijing target is located northwest of Henan, and the Chinese Mainland target tends to include Henan and Tibet. Therefore, in this study, we aimed to adjust the target detection probability to improve performance on targets with lower detection accuracy by integrating the spatial and co-existence relationships of map targets with higher detection accuracy. This paper is structured as follows: First, we introduce the map images of four targets, i.e., the Chinese Mainland, Tibet, Taiwan and Henan. Then, we detail the proposed methodology for detection probability adjustment with spatial and co-existence relationships, together with the test and evaluation results for the Chinese Mainland and Henan targets. We next report visual and quantitative evaluations of the proposed method. This is followed by a discussion and concluding remarks.

2. Data Preparation

In this study, we took four administrative regions in map images as the study targets, including the Chinese Mainland, Tibet, Taiwan and Henan, as shown in Figure 1. We manually annotated the targets with a web-based annotation tool by drawing bounding boxes to contain all the pixels of each target; the annotation included the directory of the map image file, the coordinates of the top left corner of the bounding box, the width and height of the bounding box and the label of the target (as shown in Table 1).

By using an orientation crawler, we derived 2725 map images from an internet image search engine by using the keywords Chinese Mainland, Taiwan, Tibet and map; of these, 2109 and 616 images were randomly assigned to training and test sets. In total, 1677 Taiwan

targets, 449 Tibet targets, 370 Chinese Mainland targets and 947 Henan targets were manually annotated for detection model training, and 585, 452, 993 and 286 targets, respectively, for model testing. The specific distribution is shown in Table 2.



Figure 1. The four study targets. Target 1 in the red bounding box: Chinese Mainland; Target 2 in the blue bounding box: Tibet; Target 3 in the yellow bounding box: Taiwan; Target 4 in the green bounding box: Henan.

Table 1. Target annotation format.

No.	File Path	bbox_x	bbox_y	Width	Height	Label
1	dataset/img_078.jpg	2	8	290	218	Chinese mainland
2	dataset/img_079.jpg	35	20	566	444	Chinese mainland
3	dataset/img_254.jpg	645	820	660	851	Taiwan
4	dataset/img_255.jpg	157	0	476	565	Taiwan
5	dataset/img_411.jpg	44	483	520	745	Tibet
6	dataset/img_412.jpg	15	220	291	365	Tibet
7	dataset/img_446.jpg	609	399	736	511	Henan
8	dataset/img_450.jpg	259	204	309	251	Henan

Table 2. Sample distribution of different targets.

Administrative Region	Training Targets	Test Targets	Total
Chinese Mainland	370	993	1363
Taiwan	1677	585	2262
Tibet	449	452	901
Henan	947	286	1233
Total	3443	2316	5759

3. Methods

The method in this study consisted of four stages: RetinaNet-based detection model training and prediction (Section 3.1), spatial relationship construction (Section 3.2), co-existence relationship construction (Section 3.3) and adjustment of target detection probability with constructed relationships (Section 3.4). We first built four RetinaNet models for the target detection of the four different administrative regions, and each image was predicted by using Single-Target Cascading detection [31] to represent whether the four targets existed. The predicted results were first assessed by estimating whether the targets co-existed by using the spatial relationships between different targets according to the

two criteria of area ratio and centroid orientation. Subsequently, the dependent and independent probabilities of co-existence relationships were used to adjust the detection probability (known as the prediction confidence score) to improve detection performance. The methodology framework is shown in Figure 2.

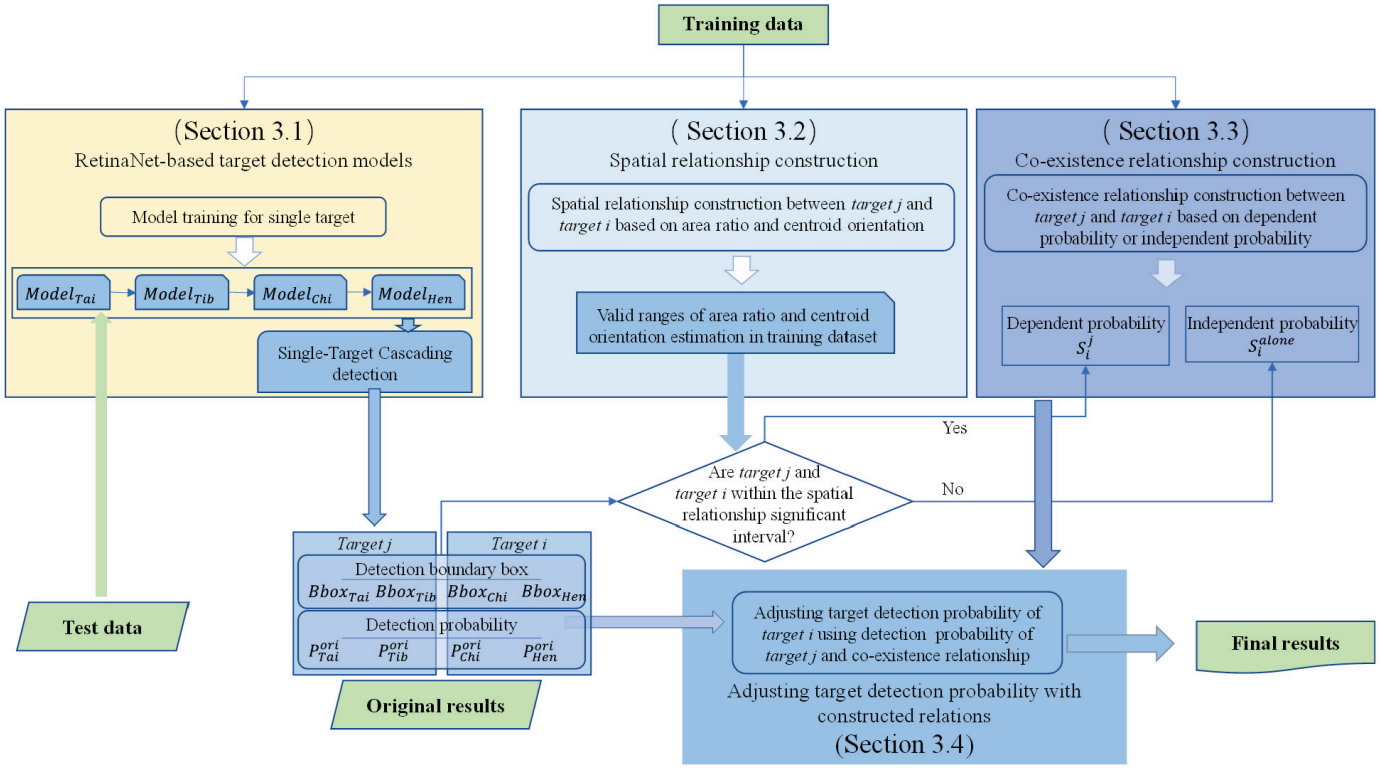


Figure 2. The methodology framework in this study.

3.1. RetinaNet-Based Target Detection Models

RetinaNet is a simple, one-stage object detector composed of two main backbone networks for generating multi-scale convolutional feature maps of the input image, i.e., ResNet50 and Feature Pyramid Network (FPN), where FPN is built on top of ResNet50. After the FPN backbone, there are two subnetworks: one for classifying anchor boxes and the other for regressing from anchor boxes to ground-truth boxes. The RetinaNet network structure used in this study is shown in Figure 3.

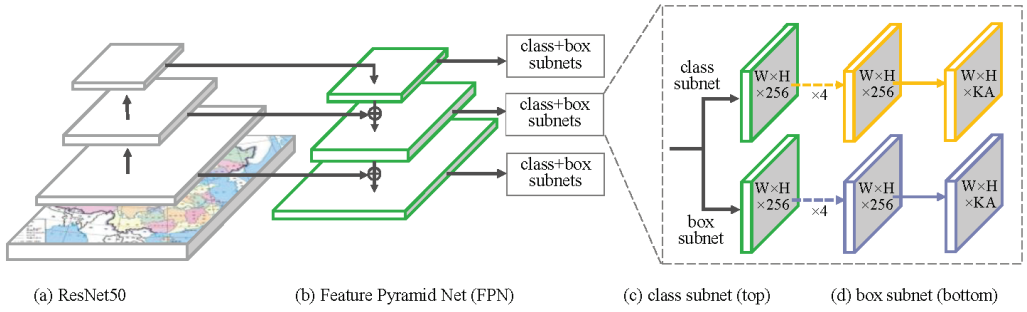


Figure 3. Basic network structure of RetinaNet.

We separately trained four basic RetinaNet models for the four map administrative region targets. The prediction strategy used in this study was Single-Target Cascading detection [31]. A full description of this detection model is beyond the scope of this study, but more information can be found in [31].

Each image was predicted by using Single-Target Cascading detection to obtain original results, indicating the presence or absence of the four targets; these represented the input for the following phase, in which they were adjusted with spatial and co-existence relationships.

3.2. Spatial Relationship Construction

The area ratio and centroid orientation were employed to measure the spatial relationships between different targets, and their valid ranges were estimated in the training samples as the statistical criteria to determine whether targets co-existed. As is well known, different targets occupy relatively stable extents on a map, which results in the relatively stable area ratio of different targets. Despite various map projections and the distortion of distance, a stable area ratio of the targets on a map is thus maintained. The area ratio between the bounding boxes of two targets in each image in the training sample set was calculated by using Equation (1), and its valid range was calculated with the 3σ rule [46,47] as the criterion to determine whether the targets co-existed.

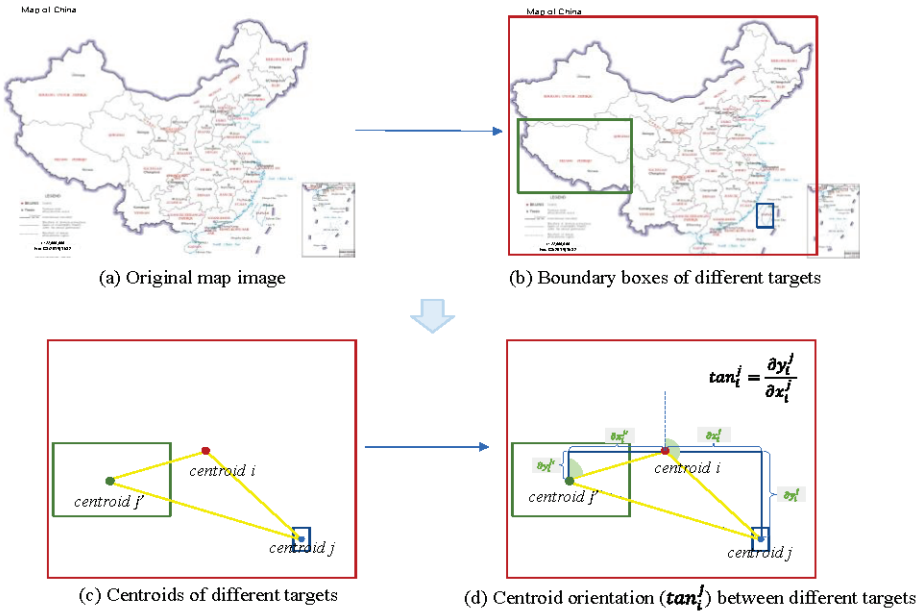
$$ratio_i^j = \frac{area_{Bbox_j}}{area_{Bbox_i}} \tag{1}$$

where $ratio_i^j$ is the area ratio between target i and target j, and $area_{Bbox_i}$ and $area_{Bbox_j}$ are the areas of the bounding boxes of targets i and j, respectively.

In addition to the area ratio above, representing the spatial extent of the proportional relationship between two targets, we also introduced centroid orientation to measure the spatial location relationship between two targets, defined as the tangent of the angle formed by the two centroids of the two targets' bounding boxes (Figure 4). Similarly to the area ratio, the valid range for the centroid orientation was obtained with the 3σ rule by calculating the tangent values for each image in the training sample, and it was used to determine whether the targets co-existed.

$$tan_i^j = \frac{\partial y_{Bbox_i}^{Bbox_j}}{\partial x_{Bbox_i}^{Bbox_j}} \tag{2}$$

where tan_i^j is the centroid orientation between target i and target j, and $\partial y_{Bbox_i}^{Bbox_j}$ and $\partial x_{Bbox_i}^{Bbox_j}$ are the y- and x-difference, respectively, between target i and target j of the centroid point.



(a) Original map image (b) Boundary boxes of different targets

(c) Centroids of different targets (d) Centroid orientation (tan_i^j) between different targets

Figure 4. Illustration of centroid orientation relationship between different targets.

In the context of predictions generated with the RetinaNet-based Single-Target Cascading model, two administrative region targets are considered to co-exist only when they concurrently satisfy the criteria of both the area ratio and centroid orientation. Otherwise, each target exists independently of the other.

3.3. Co-Existence Relationship Construction

Spatial relationships were used to automatically confirm whether two targets co-existed or not. The dependent probability of target co-existence and the independent probability of the target were also calculated in the training samples to further estimate the probability of co-existence, which, in turn, was used to improve the Single-Target Cascading model detection results. Specifically, based on all training samples with target labels, there were seven co-existence patterns (Table 3). The dependent and independent probabilities were calculated with Equations (3) and (4), where Equation (3) represents the probability of target *j* co-existing with target *i* in a map image.

Table 3. Co-existence patterns of different targets in the training samples (where 1 indicates the existence and 0 the non-existence of a target in an image).

Co-Existence Pattern \ Administrative Region	Chinese Mainland (or Henan)	Taiwan	Tibet
1	1	1	1
2	1	0	1
3	1	0	0
4	0	1	0
5	1	1	0
6	0	1	1
7	0	0	1

The following dependent and independent probability formulas were derived from the co-existence patterns shown in Table 3.

$$S_i^j = \frac{num_i^j}{num_i} \tag{3}$$

$$S_i^{alone} = \frac{num_i^0}{num_{all}} \tag{4}$$

where S_i^j represents the existence probability of target *j* when target *i* exists in a map image, i.e., the dependent probability for target *j* with respect to target *i*; num_j represents the number of images containing target *j*; num_i^j represents the number of images containing both target *i* and target *j*; S_i^{alone} represents the independent probability that an image solely contains target *i* without including other targets; num_{all} represents the total number of training images; and num_i^0 represents the number of sample images that exclusively contain target *i*.

3.4. Adjustment of Target Detection Probability with Constructed Relations

We aimed to improve one target's (hereafter referred to as target *i*) detection result with the commission error (see Section 4.4) by using the dependent probability of target co-existence and the independent probability of the target. When the co-existence of two or three targets is confirmed by a spatial relationship, the detection probability of target *i* can be adjusted with the detection results of the other targets in the same image. Specifically, the detection probability of target *i* is adjusted by using the original result of target *j*'s detection probability (known as the prediction confidence score) multiplied by the dependent probability of targets *j* and *i* if the latter co-exist in an image (Equation (5)). If target *i* exists independently of other targets, its detection probability is adjusted by using its original detection probability multiplied by its independent probability (Equation (6)).

The final detection probability of target i is the maximum of all its adjusted detection probability values (Equation (7)).

$$P_i^{j_Sup} = P_j^{ori} \times S_j^i \tag{5}$$

$$P_i^{alone} = P_i^{ori} \times S_i^{alone} \tag{6}$$

$$P_i^{final} = \max\left(P_j^{i_Sup},\ P_i^{alone}\right) \tag{7}$$

where P_j^{ori} is the prediction confidence score of target j determined with the Single-Target Cascading model; S_j^i describes the existence probability of target i when target j exists; $P_i^{j_Sup}$ is the prediction confidence score of target j to which the target i detection result has been added; P_i^{ori} is the prediction confidence of target i determined with its detection model; S_i^{alone} describes the probability of target i existing independently of other targets; P_i^{alone} is the prediction confidence score of target i existing alone; $P_i^{j_Sup}$ represents the corrected prediction confidence of target i, which is based on the co-existence-dependent probability of target i with respect to j; P_i^{alone} represents the confidence score of target i existing alone; and P_i^{final} is the maximum value among $P_i^{j_Sup}$ and P_i^{alone} and represents the final prediction confidence score of target i with different adjusted targets. The value of P_i^{ori} is 0 when target i does not exist in the image.

4. Experiment and Results

The performance of the proposed methods was evaluated with precision, recall and the *F1* score based on the test data. Precision measures the accuracy of the detected targets, and recall measures the integrity of detection. The *F1* score is a combination of precision and recall and provides a more comprehensive basis to evaluate a model.

$$precision = \frac{TP}{TP + FP} \tag{8}$$

$$recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1_{score} = \frac{2 \times precision \times recall}{precision + recall} \tag{10}$$

where *TP* represents the number of correctly predicted targets, *FP* represents the incorrectly predicted results, and *FN* represents the number of targets that have been missed in target detection.

The *IOU* (intersection over union) was used to evaluate how well the predicted bounding boxes matched the ground-truth data (Equation (11)). The predicted bounding boxes were considered to be correct when the IOU reached an empirical threshold of 0.5. We used the *F1* score with an *IOU* threshold of 0.5 as the final evaluation indicator of the model, where B_p is the predicted bounding box and B_{gt} is the hand-labeled bounding box of a target.

$$IOU = \frac{area\left(B_{gt} \cap B_p\right)}{area\left(B_{gt} \cup B_p\right)} \tag{11}$$

4.1. Original Results of RetinaNet-Based Target Detection

Figure 5 shows the original results of the four targets obtained by using Single-Target Cascading detection. The Taiwan target showed the best performance, with the highest *F1* score value of 0.94, while the Henan target had the lowest *F1* score value of 0.75. The precision-based evaluation results of the detection results of the Henan and Chinese Mainland targets were significantly lower (only 0.64 and 0.62) than those of the other two targets. All the metrics of the Taiwan and Tibet target detection results are desirable. Therefore, in this study, we used the original detection results of the Taiwan and Tibet targets to adjust those of the Henan and Chinese Mainland targets with spatial and co-existence relationships.

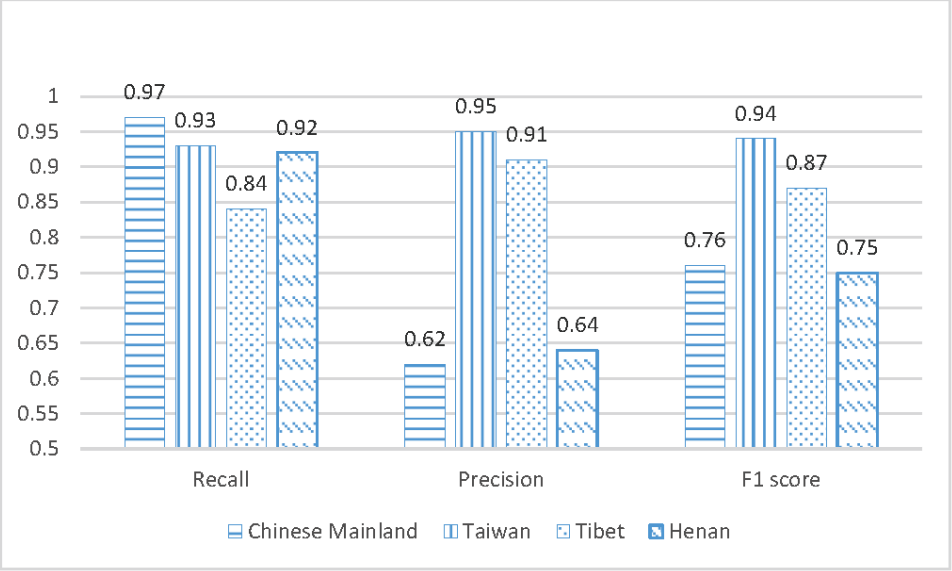


Figure 5. A comparison of the evaluation metrics of the four targets' original detection results.

4.2. Spatial Relationship between Targets

(1) Area ratio

$ratio_{Chi}^{Tib} = \frac{area_{Bbox_Tib}}{area_{Bbox_Chi}}$ and $ratio_{Chi}^{Tai} = \frac{area_{Bbox_Tai}}{area_{Bbox_Chi}}$ were used to measure the area ratios of the Chinese Mainland–Tibet and Chinese Mainland–Taiwan targets according to Equation (1). In parallel, $ratio_{Hen}^{Tib} = \frac{area_{Bbox_Tib}}{area_{Bbox_Hen}}$ and $ratio_{Hen}^{Tai} = \frac{area_{Bbox_Tai}}{area_{Bbox_Hen}}$ were used to measure those of the Henan–Tibet and Henan–Taiwan targets.

Table 4 shows that the area ratio of the Chinese Mainland–Taiwan targets in the training dataset ranged from 0.002 to 0.020, with a significant interval (mean ± 3σ) from 0 to 0.0114, while that of the Chinese Mainland–Tibet targets ranged from 0.0706 to 0.169, with a significant interval (mean ± 3σ) from 0.0753 to 0.167. We also plotted the area ratio distributions for the Chinese Mainland and Taiwan targets, which were found to be normal, with a μ of 0.00528 and a σ of 0.00203; a similar distribution was observed for the Chinese Mainland and Tibet targets, with a μ of 0.121 and a σ of 0.0153. It was evident that the two normal distributions were separable (Figure 6). This can be used to determine whether the area ratio between two targets is reliable; that is to say, two targets probably co-exist if their area ratio is within the possible range, and vice versa. Table 5 shows the area ratio distribution for the Henan–Taiwan and Henan–Tibet targets in the training dataset, and Figure 7 shows a histogram of the area ratio distributions for the Henan–Taiwan and Henan–Tibet targets.

Table 4. Area ratio distributions for Chinese Mainland–Taiwan ($ratio_{Chi}^{Tai}$) and Chinese Mainland–Tibet ($ratio_{Chi}^{Tib}$) targets in training dataset.

Area Ratio ($ratio_i^j$)	Min	Max	Mean	σ	Mean − 3σ	Mean + 3σ
$ratio_{Chi}^{Tai}$	0.002	0.020	0.00528	0.00203	−0.00081 (set 0)	0.0114
$ratio_{Chi}^{Tib}$	0.0706	0.169	0.121	0.0153	0.0753	0.167

Table 5. Area ratio distributions for Henan–Taiwan ($ratio_{Hen}^{Tai}$) and Henan–Tibet ($ratio_{Hen}^{Tib}$) targets in training dataset.

Area Ratio ($ratio_i^j$)	Min	Max	Mean	σ	Mean − 3σ	Mean + 3σ
$ratio_{Hen}^{Tai}$	0.0963	0.756	0.250	0.0638	0.0586	0.441
$ratio_{Hen}^{Tib}$	4.723	8.867	6.195	0.659	4.218	8.172

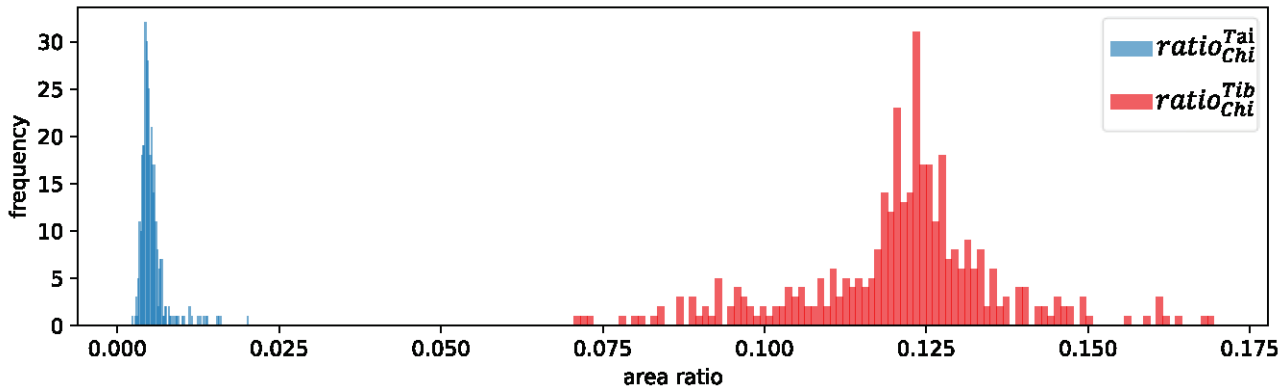


Figure 6. Histogram of area ratio distributions for Chinese Mainland–Taiwan (blue) and Chinese Mainland–Tibet (red) targets.

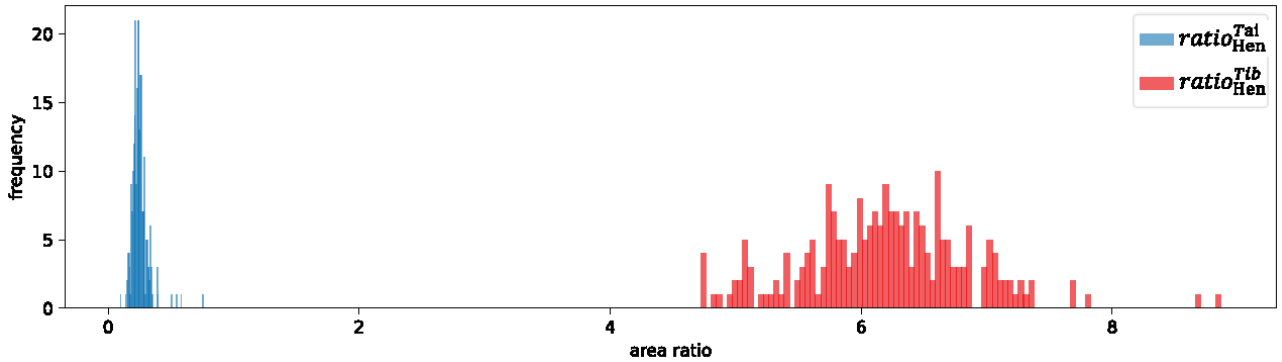


Figure 7. Histogram of area ratio distributions for Henan–Taiwan (blue) and Henan–Tibet (red) targets.

(2) Centroid orientation

$tan_{Chi}^{Tib} = \frac{\partial y_{Bbox_Chi}^{Bbox_Tib}}{\partial x_{Bbox_Chi}^{Bbox_Tib}}$ and $tan_{Chi}^{Tai} = \frac{\partial y_{Bbox_Chi}^{Bbox_Tai}}{\partial x_{Bbox_Chi}^{Bbox_Tai}}$ were employed to measure the centroid orientation of the Chinese Mainland–Tibet and Chinese Mainland–Taiwan targets in the training dataset according to Equation (2). In parallel, $tan_{Hen}^{Tib} = \frac{\partial y_{Bbox_Hen}^{Bbox_Tib}}{\partial x_{Bbox_Hen}^{Bbox_Tib}}$ and $tan_{Hen}^{Tai} = \frac{\partial y_{Bbox_Hen}^{Bbox_Tai}}{\partial x_{Bbox_Hen}^{Bbox_Tai}}$ were employed to measure that of the Henan–Tibet and Henan–Taiwan targets.

Table 6 shows that the centroid orientation of the Chinese Mainland–Taiwan targets in the training maps ranged from 0.404 to 1.413, with a significant interval (mean $\pm$ 3σ) from 0.393 to 1.119, while that of the Chinese Mainland–Tibet targets ranged from -1 to 0.233, with a significant interval (mean $\pm$ 3σ) from -0.753 to 0.0988. We also plotted the centroid orientation distributions of the Chinese Mainland and Taiwan targets, which were found to be normal, with a μ of 0.756 and a σ of 0.121; a similar distribution was observed for the Chinese Mainland and Tibet targets, with a μ of -0.327 and a σ of 0.142. It was evident that the two normal distributions were separable (Figure 8). Similarly to the area ratio, this can be used to determine whether the centroid orientation between two targets is reliable; that is to say, two targets probably co-exist if the centroid orientation is within the possible range, and vice versa. Table 7 shows the centroid orientation distribution parameters and interval estimates for the Henan–Taiwan and Henan–Tibet targets in the training dataset, and Figure 9 shows a histogram of the centroid orientation distributions for the Henan–Taiwan and Henan–Tibet targets.

The valid ranges for the area ratio and centroid orientation were determined by using the significant intervals above. For target prediction based on the Single-Target Cascading model, only when two administrative region targets simultaneously fall within the valid ranges (i.e., significant intervals) of area ratio and centroid orientation are they considered co-existent; otherwise, each target exists independently of the other target.

Table 6. Centroid orientation distribution parameters and interval estimates for Chinese Mainland–Taiwan (tan^{Tai}_{Chi}) and Chinese Mainland–Tibet (tan^{Tib}_{Chi}) targets in training dataset.

Centroid Orientation (tan^{j}_{i})	Min	Max	Mean	σ	Mean − 3σ	Mean + 3σ
tan^{Tai}_{Chi}	0.404	1.413	0.756	0.121	0.393	1.119
tan^{Tib}_{Chi}	−1	0.233	−0.327	0.142	−0.753	0.0988

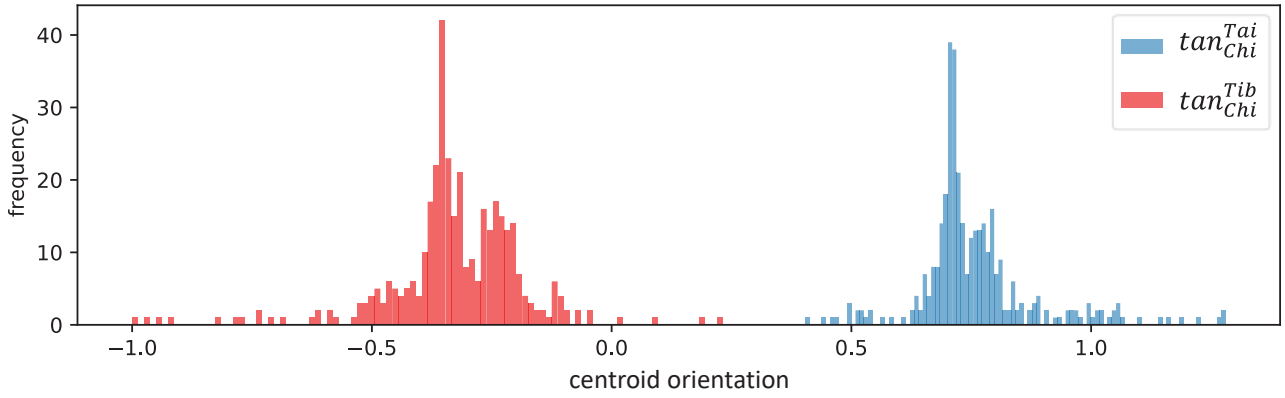


Figure 8. Histogram of centroid orientation distributions for Chinese Mainland–Taiwan (blue) and Chinese Mainland–Tibet (red) targets.

Table 7. Centroid orientation distribution parameters and interval estimates for Henan–Taiwan (tan^{Tai}_{Hen}) and Henan–Tibet (tan^{Tib}_{Hen}) targets in training dataset.

Centroid Orientation (tan^{j}_{i})	Min	Max	Mean	σ	Mean − 3σ	Mean + 3σ
tan^{Tai}_{Hen}	0.810	1.958	1.328	0.148	0.884	1.772
tan^{Tib}_{Hen}	−0.136	0.207	−0.0613	0.0409	−0.184	0.0614

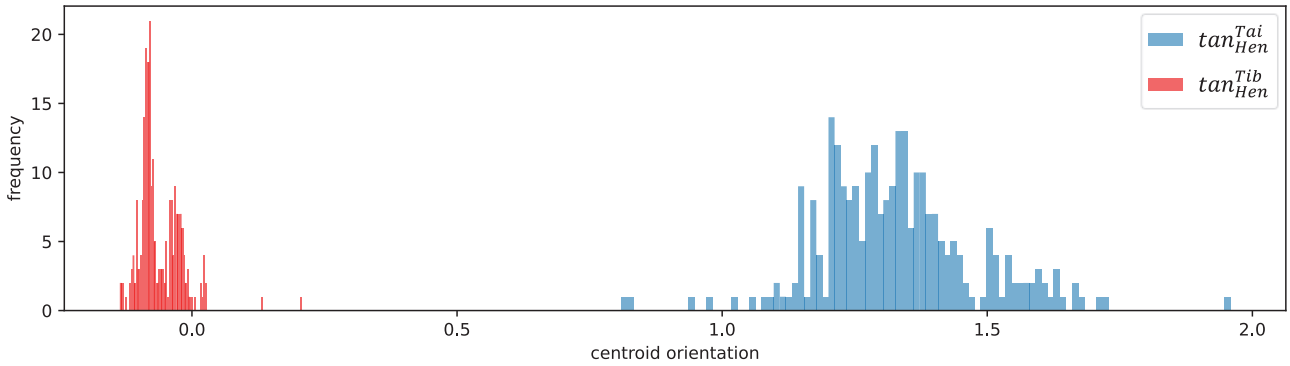


Figure 9. Histogram of centroid orientation distributions for Henan–Taiwan (blue) and Henan–Tibet (red) targets.

4.3. Co-Existence Relationship Construction for Chinese Mainland Target

Table 8 shows the number of different co-existence patterns in the training dataset.

Table 8. Distribution of co-existence patterns in training samples.

Co-Existence Pattern	1	2	3	4	5	6	7
Number (Chinese Mainland)	1305	28	3	585	3	5	180
Number (Henan)	839	7	0	506	82	471	201

Table 9 lists the dependent probabilities for different co-existence patterns of the Chinese Mainland–Tibet, Chinese Mainland–Taiwan, Henan–Tibet and Henan–Taiwan targets, determined according to Equation (3), as well as the independent probability of the Chinese Mainland and Henan targets existing alone, determined according to Equation (4).

Table 9. Dependent and independent probabilities for different targets and corresponding co-existence patterns.

Co-Existence Pattern	Probability Calculation	Value
Chinese Mainland–Tibet	$S_{TIb}^{Chi} = \frac{num_{Tib}^{Chi}}{num_{Tib}} = \frac{1305+28}{1305+28+5+180}$	0.88
Chinese Mainland–Taiwan	$S_{Tai}^{Chi} = \frac{num_{Tai}^{Chi}}{num_{Tai}} = \frac{1305+3}{1305+585+3+5}$	0.69
Chinese Mainland alone	$S_{Chi}^{alone} = \frac{num_{Chi}^{0}}{num_{all}} = \frac{3}{2109}$	0.0014
Henan–Tibet	$S_{TIb}^{Hen} = \frac{num_{Tib}^{Hen}}{num_{Tib}} = \frac{839+7}{839+7+471+210}$	0.554
Henan–Taiwan	$S_{Tai}^{Hen} = \frac{num_{Tai}^{Hen}}{num_{Tai}} = \frac{839+82}{839+506+82+471}$	0.485
Henan alone	$S_{Hen}^{alone} = \frac{num_{Hen}^{0}}{num_{all}} = \frac{0}{2106}$	0

4.4. Adjusted Detection Results for Chinese Mainland Target

Table 10 shows the original prediction confidence score adjustment and final prediction confidence calculation according to Equations (5)–(7).

Table 10. Prediction confidence adjustment for Chinese Mainland and Henan targets.

Co-Existence Pattern	Confidence Score Adjustment Formula
Chinese Mainland–Taiwan	$P_{Chi}^{Tai_Sup} = P_{Tai}^{ori} \times S_{Tai}^{Chi}$
Chinese Mainland–Tibet	$P_{Chi}^{Tib_Sup} = P_{Tib}^{ori} \times S_{Tib}^{Chi}$
Chinese Mainland alone	$P_{Chi}^{alone} = P_{Chi}^{ori} \times S_{Chi}^{alone}$
Final detection probability	$P_{Chi}^{final} = max\left(P_{Chi}^{TaiSup}, P_{Chi}^{TibSup}, P_{Chi}^{alone}\right)$
Henan–Taiwan	$P_{Hen}^{Tai_Sup} = P_{Tai}^{ori} \times S_{Tai}^{Hen}$
Henan–Tibet	$P_{Hen}^{Tib_Sup} = P_{Tib}^{ori} \times S_{Tib}^{Hen}$
Henan alone	$P_{Hen}^{alone} = P_{Hen}^{ori} \times S_{Hen}^{alone}$
Final detection probability	$P_{Hen}^{final} = max\left(P_{Hen}^{TaiSup}, P_{Hen}^{TibSup}, P_{Hen}^{alone}\right)$

Figure 10 shows the detection results for the Chinese Mainland before (a) (incorrect results) and after (b) (correct results) their adjustment with spatial and co-existence relationships; their comparison indicates that the prediction confidence score of the Chinese Mainland target was decreased by applying spatial and co-existence relationships. For example, in the first row of Figure 10, the prediction confidence score values are 0.77 and 0.0011 before and after adjustment, respectively. This visually confirmed that the incorrect detection result could be successfully eliminated, which improved the precision score of the Chinese Mainland target detection results. Similarly, the incorrect detection results of the Henan target could be successfully eliminated, as shown in Figure 11.

Figure 12 shows that some test images with correct Chinese Mainland target detection results were mistakenly removed after adjustment.

Figure 13 shows that some test images with correct Henan target detection results were mistakenly removed after the adjustment.

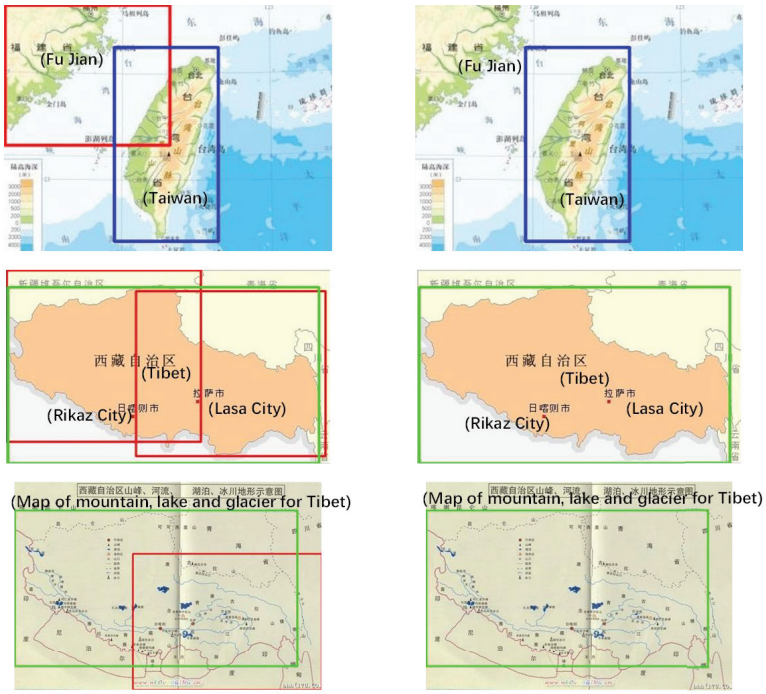


Figure 10. Incorrect target detection results for Chinese Mainland target before (**a**) and after (**b**) adjustment with spatial and co-existence relationships. Red box: Chinese Mainland target; green box: Tibet target; blue box: Taiwan target.

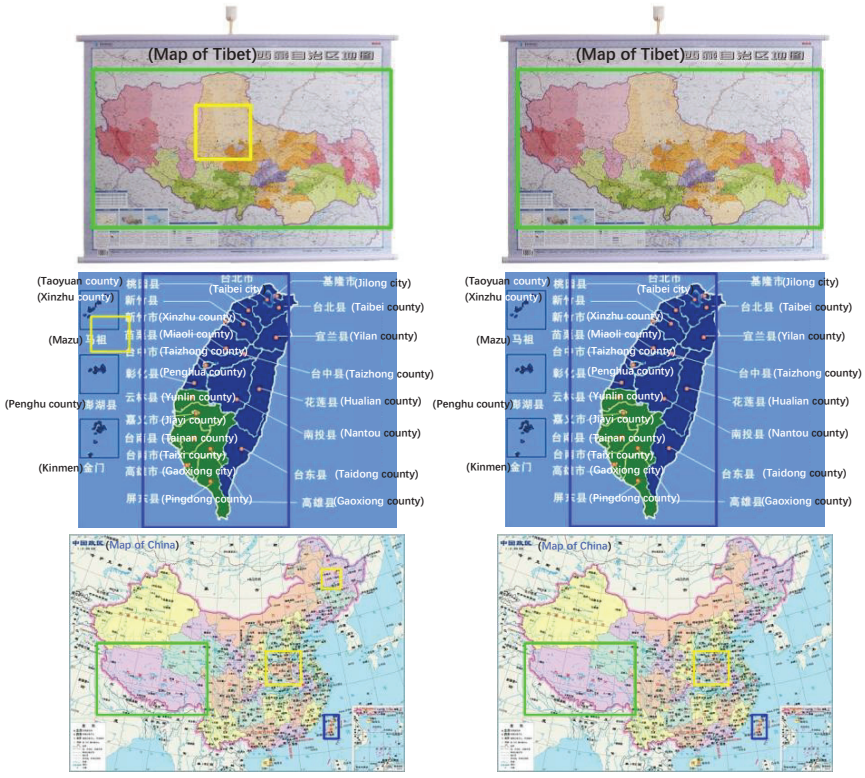
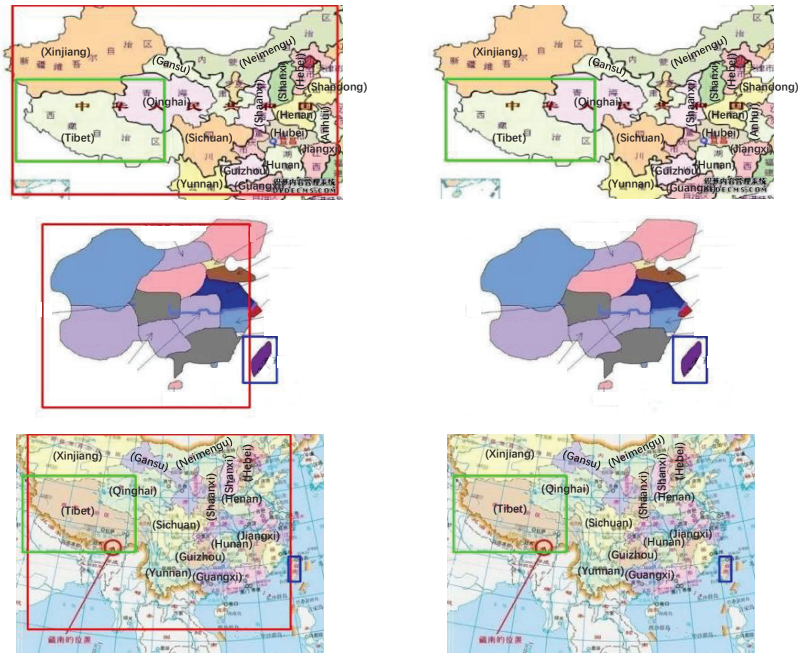
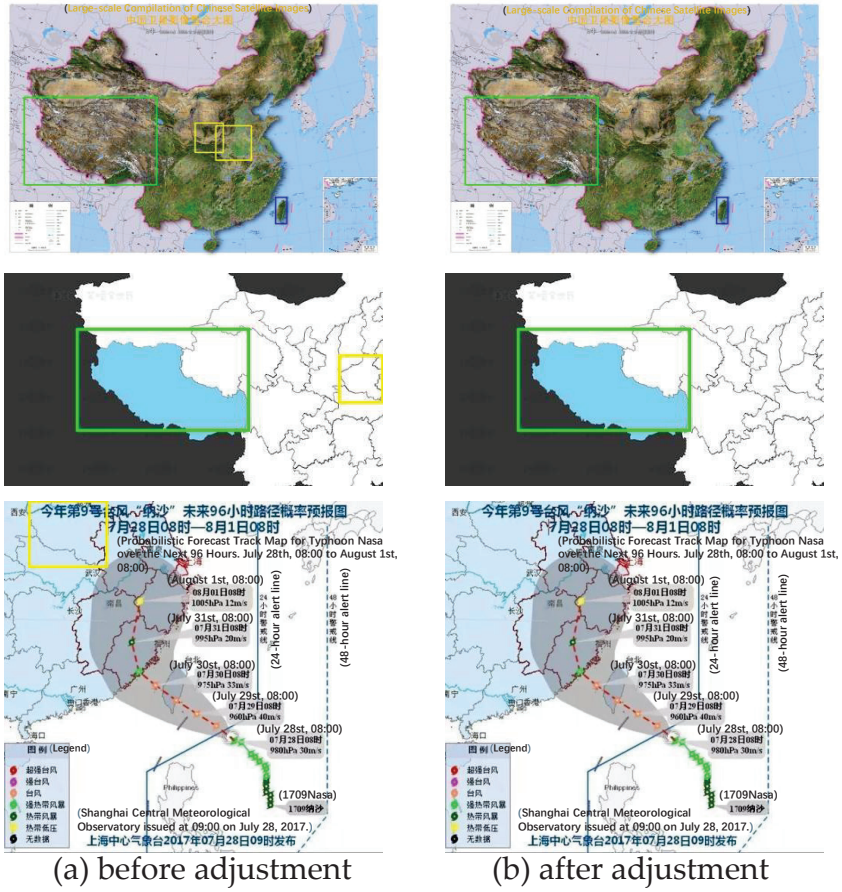


Figure 11. Incorrect target detection results for Henan target before (**a**) and after (**b**) adjustment with spatial and co-existence relationships. Yellow box: Henan target; green box: Tibet target; blue box: Taiwan target.



(a) before adjustment (b) after adjustment

Figure 12. Chinese Mainland target's correct detection results (**a**) were removed after (**b**) adjustment with spatial and co-existence relationships. Red box: Chinese Mainland target; green box: Tibet target; blue box: Taiwan target.



(a) before adjustment (b) after adjustment

Figure 13. Henan target's correct detection results (**a**) were removed after (**b**) adjustment with spatial and co-existence relationships. Yellow box: Henan target; green box: Tibet target; blue box: Taiwan target.

Figure 14 shows a comparison of the map detection accuracy rates for the Chinese Mainland target before and after the adjustment of the prediction probability. Following prediction confidence score correction, the detection of the Chinese Mainland target in the same test dataset showed significant improvement, with an R, P and *F1* score of 0.82, 0.96 and 0.88, respectively. The probability adjustment resulted in a 12% increment in the *F1* score, even though the recall value decreased; precision increased by 34% to a value of 0.96, which indicates that the commission error of the Chinese Mainland target was significantly eliminated. Regarding the Henan target, the *F1* score increased by 2% after the adjustment of the prediction probability, as shown in Figure 15.

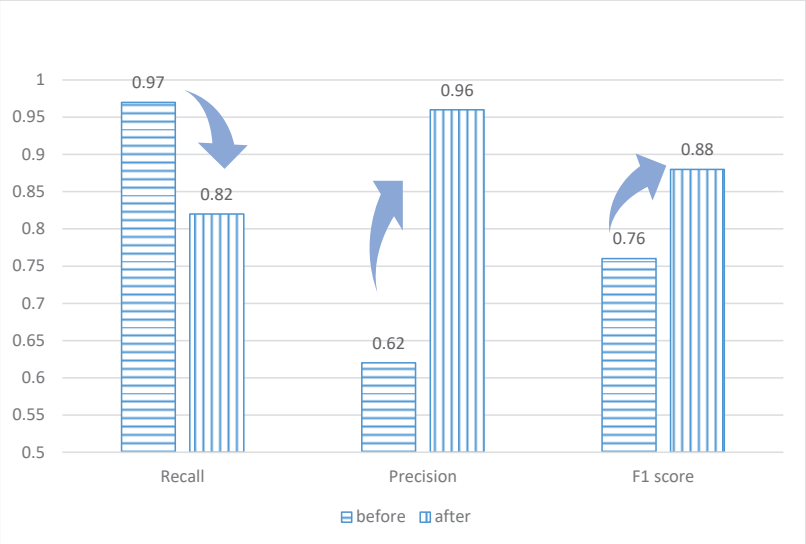


Figure 14. Changes in evaluation metrics for Chinese Mainland target after adjustment.

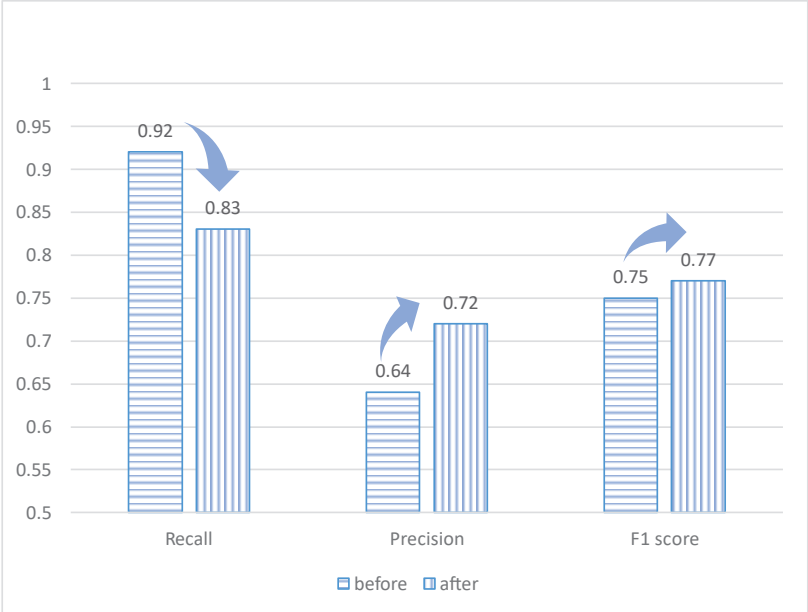


Figure 15. Changes in evaluation metrics for Henan target after adjustment.

5. Discussion

5.1. Improvement in Precision and F1 Score of Target Detection Results Following Adjustment

The Single-Target Cascading detection model based on RetinaNet identifies targets in images by extracting the color and texture features of objects; however, it is challenging to integrate the relationships between targets into the model. Previous studies have

rarely taken the relationships between targets into account. However, maps, as carriers of geographic information, display very strong spatial correlations among different administrative region targets.

In this study, we integrated the spatial and co-existence relationships between the targets with higher accuracy (i.e., Taiwan and Tibet) to optimize the model detection results of the targets with lower accuracy (i.e., Chinese Mainland and Henan), based on the original target detection results. The area ratio and centroid orientation were used to measure the spatial relationships, which were used to determine whether two targets co-existed or not. Moreover, we obtained the dependent and independent probabilities of the Chinese Mainland and Henan targets by calculating the distribution of the co-existence patterns and used them to measure the co-existence relationships between different targets in a map; then, they were used to adjust the prediction confidence scores of the Chinese Mainland and Henan targets. Thus, the relationships between different targets act as a tool to improve the target detection model's accuracy.

The results of this study clearly show that the spatial and co-existence relationships between the Chinese Mainland and Taiwan, as well as those between the Chinese Mainland and Tibet, significantly contributed to the improvement in detection accuracy for the Chinese Mainland target. The Chinese Mainland target detection results were adjusted by using the high-precision model (Taiwan and Tibet) with these relationships to obtain a 34% improvement in precision and a 12% improvement in *F1* score, which indicate strong usability. Regarding the Henan target, the detection results showed an 8% improvement in precision and a 2% improvement in *F1* score. We noticed that the area ratio distribution for the Henan–Tibet ($ratio_{Hen}^{Tib}$) targets had a high σ value (0.659), which means that it was not stable and directly led to poor precision and *F1* score adjustment.

5.2. Loss of Recall in Target Detection Results Following Adjustment

A small portion of the correct Chinese Mainland and Henan target detection results were erroneously removed after the adjustment, as shown in Figures 12 and 13, because the area ratio or centroid orientation fell outside the valid range. This erroneous removal of targets reduced the recall score (see Figures 14 and 15) of the Chinese Mainland and Henan target detection results.

The Chinese Mainland and Henan targets' prediction confidence scores were decreased by S_{Chi}^{alone} and S_{Hen}^{alone}, respectively, if the Taiwan and Tibet targets were not identified by the models in the same map, which means that the recall loss for other targets accumulated in that for the Chinese Mainland and Henan targets. However, the valid ranges for the area ratio and centroid orientation could also affect the recall score for the Chinese Mainland and Henan targets. In future research, we will calculate the significant intervals from a growing dataset or infer them from input images with a network rather than obtaining them by performing statistical analysis on a small, manually annotated, fixed training dataset.

5.3. Limitations of Spatial Relationship Measurement

There are also some limitations to the proposed adjustment method. The valid ranges for the area ratio and centroid orientation are essential to spatial relationship construction. Moreover, they can also easily influence co-existence determination. In this study, we derived the valid ranges from the statistical analysis of the training samples with the simple 3σ rule [43,44]. The distribution of the training samples was biased relative to the overall image data. Therefore, comprehensive insights into the overall data distribution remain elusive due to the limitations of our sample dataset, and the generalization of the valid ranges acquired in this study appears to be limited. Moreover, the bias of the valid ranges decreased the recall score of the Chinese Mainland target's detection results in this study. Spatial and co-existence relationship patterns should be explored with more advanced methods, such as dot-product attention [48], to investigate their complex nature.

6. Conclusions

In contrast to existing research focused on optimizing the structure of object detection models, in this study, we established a method to improve target detection results with low accuracy by adjusting those with higher accuracy with spatial and co-existence relationships.

In this study, the Chinese Mainland, Taiwan, Tibet and Henan were taken as target examples, and the Single-Target Cascading model was applied to map images for original target detection. At the same time, spatial and co-existence relationships were used to improve target detection precision for the Chinese Mainland and Henan targets. We used the area ratio and centroid orientation to measure the spatial relationship between two targets and the dependent and independent probabilities of different targets to measure their co-existence relationships.

In this study, we used the relationships between different targets to decrease the recall score and improve precision for the detection of the Chinese Mainland and Henan targets, thus balancing the two metrics.

The results demonstrate that the use of spatial and co-existence relationships between different administrative regions can significantly improve overall detection accuracy by making use of the high-accuracy advantage of one model to improve another. In this work, the Chinese Mainland and Henan targets, which had smaller training datasets, obtained higher *F1* score values after the detection result adjustment. In conclusion, the relationships between different targets in a map are valuable, as they can reduce the labor-intensive nature of training data preparation. The results of this study provide a basis for improving the efficiency of map search and intelligent map interpretation.

Author Contributions: Conceptualization, Yong Wang, Fu Ren and Kaixuan Du; methodology, Kaixuan Du; validation, Xianghong Che and Kaixuan Du; formal analysis, Xianghong Che and Jiping Liu; data curation, Yong Wang, Kaixuan Du, Jiaxin Hou and Zewei You; writing—original draft preparation, Kaixuan Du; writing—review and editing, Kaixuan Du, Xianghong Che and Zewei You; visualization, Kaixuan Du; supervision, Fu Ren; project administration, Yong Wang; funding acquisition, Yong Wang and Fu Ren. All authors have read and agreed to the published version of the manuscript.

Funding: This research study was funded by the National Key Research and Development Program of China [grant number 2022YFC3005704], Basic Scientific Research Foundation of Chinese Academy of Surveying and Mapping [grant number AR2205] and Special Business Expenses of the Ministry of Natural Resources [grant number 121136000000180004].

Data Availability Statement: The data used in this study are publicly available from multiple public websites. The map images used in this study can be downloaded from the following websites: bzdt.ch.mnr.gov.cn (accessed on 27 November 2021), www.photophoto.cn (accessed on 27 November 2021), www.ce.cn (accessed on 27 November 2021), www.gov.cn (accessed on 27 November 2021), www.cnr.cn (accessed on 27 November 2021), etc.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, J. Cartography: From Digital to Intelligent. *Geomat. Inf. Sci. Wuhan Univ.* **2022**, *47*, 1963–1977.
2. Hanson, P. Administrative Regions and the Economy. In *The Legacy of the Soviet Union*; Springer: London, UK, 2004; pp. 144–168.
3. Andersen, A.K. Are commuting areas relevant for the delimitation of administrative regions in Denmark? *Reg. Stud.* **2002**, *36*, 833–844. [CrossRef]
4. Kuzdeuov, A.; Baimukashev, D.; Karabay, A.; Ibragimov, B.; Mirzakhmetov, A.; Nurpeiissov, M.; Lewis, M.; Varol, H.A. A network-based stochastic epidemic simulator: Controlling COVID-19 with region-specific policies. *IEEE J. Biomed. Health* **2020**, *24*, 2743–2754. [CrossRef]
5. Lemieux, F. The impact of a natural disaster on altruistic behaviour and crime. *Disasters* **2014**, *38*, 483–499. [CrossRef]
6. Li, H.; Kopsakangas-Savolainen, M.; Yan, M.; Wang, J.; Xie, B. Which provincial administrative regions in China should reduce their coal consumption? An environmental energy input requirement function based analysis. *Energy Policy* **2019**, *127*, 51–63. [CrossRef]
7. Lin, Y.; Huang, J.; Li, M.; Lin, R. Does lower regional density result in less CO_2 emission per capita? Evidence from prefecture-level administrative regions in China. *Environ. Sci. Pollut. Res.* **2022**, *29*, 29887–29903. [CrossRef]

8. Trappey, A.J.; Trappey, C.V.; Lin, G.Y.; Chang, Y. The analysis of renewable energy policies for the Taiwan Penghu island administrative region. *Renew. Sustain. Energy Rev.* **2012**, *16*, 958–965. [CrossRef]
9. Wang, K.; Wang, F. Theory and measurement model of administrative region potential from a perspective of administrative division adjustment: Taking Chongqing city as a case study. *J. Geogr. Sci.* **2020**, *30*, 1341–1362. [CrossRef]
10. Wang, Y. Multiperiod optimal allocation of emergency resources in support of cross-regional disaster sustainable rescue. *Int. J. Disaster Risk Sci.* **2021**, *12*, 394–409. [CrossRef]
11. Duan, W.; Chiang, Y.; Leyk, S.; Uhl, J.H.; Knoblock, C.A. Automatic alignment of contemporary vector data and georeferenced historical maps using reinforcement learning. *Int. J. Geogr. Inf. Sci.* **2020**, *34*, 824–849. [CrossRef]
12. Kang, Y.; Gao, S.; Roth, R.E. Artificial intelligence studies in cartography: A review and synthesis of methods, applications, and ethics. *Cartogr. Geogr. Inf. Sci.* **2024**, 1–32. [CrossRef]
13. Uhl, J.H.; Leyk, S.; Li, Z.; Duan, W.; Shbita, B.; Chiang, Y.; Knoblock, C.A. Combining remote-sensing-derived data and historical maps for long-term back-casting of urban extents. *Remote Sens.* **2021**, *13*, 3672. [CrossRef]
14. Saini, A.; Mantosh, B. Object Detection in Underwater Image by Detecting Edges using Adaptive Thresholding. In Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019; pp. 628–632.
15. Raja, R.; Kumar, S.; Mahmood, M.R. Color object detection based image retrieval using ROI segmentation with multi-feature method. *Wirel. Pers. Commun.* **2020**, *112*, 169–192. [CrossRef]
16. Lienhart, R.; Maydt, J. An extended set of haar-like features for rapid object detection. In Proceedings of the 2002 International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002; pp. 900–903.
17. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
18. Felzenszwalb, P.; McAllester, D.; Ramanan, D. A discriminatively trained, multiscale, deformable part model. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
19. Hsu, C.W.; Lin, C.J. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 415–425. [CrossRef]
20. Schneiderman, H.; Kanade, T. Object detection using the statistics of parts. *Int. J. Comput. Vision* **2004**, *56*, 151–177. [CrossRef]
21. Prasad, D.K. Survey of the problem of object detection in real images. *Int. J. Image Process.* **2012**, *6*, 441.
22. Lu, D.; Weng, Q. A survey of image classification methods and techniques for improving classification performance. *Int. J. Remote Sens.* **2007**, *28*, 823–870. [CrossRef]
23. Zhang, K.; Feng, X.H.; Guo, Y.R.; Su, Y.K.; Zhao, K.; Zhao, Z.B.; Ma, Z.Y.; Ding, Q.L. Overview of deep convolutional neural networks for image classification. *J. Image Graph.* **2021**, *26*, 2305–2325.
24. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
25. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal.* **2017**, *40*, 834–848. [CrossRef]
26. Bai, Y.; Liu, J.; Huang, L.; Bai, J.; Che, X. Analysis of two convolutional neural networks for map image recognition. *Sci. Surv. Mapp.* **2021**, *46*, 126–134.
27. Cui, T.; Liu, J.; Luo, A. Intelligent identification method of network map images based on convolutional neural network. *Sci. Surv. Mapp.* **2019**, *44*, 118–123.
28. Du, K.; Wang, L.; Wang, Y.; Che, X. An identification method of map images based on activate learning and convolutional neural network. *Sci. Surv. Mapp.* **2020**, *45*, 139–147.
29. Wang, Z.; Liu, J.; Che, X.; Wang, Y.; Du, K. Research on map similarity matching method based on convolutional neural network. *Sci. Surv. Mapp.* **2022**, *47*, 169–175.
30. Zhou, X.; Li, W.; Arundel, S.T.; Liu, J. Deep convolutional neural networks for map-type classification. *arXiv* **2018**, arXiv:1805.10402.
31. Du, K.; Che, X.; Wang, Y.; Liu, J.; Luo, A.; Ma, R.; Xu, S. Comparison of RetinaNet-Based Single-Target Cascading and Multi-Target Detection Models for Administrative Regions in Network Map Pictures. *Sensors* **2022**, *22*, 7594. [CrossRef] [PubMed]
32. Ren, J.; Liu, W.; Li, Z.; Li, R.; Zhai, X. Intelligent detection of "Problematic Map" using convolutional neural network. *Geomat. Inf. Sci. Wuhan Univ.* **2021**, *46*, 570–577.
33. Wang, Z.; Fu, X.; Du, K.; Liu, J.; Che, X. Detection of typical geographic object in maps based on deep learning. *Bull. Surv. Mapp.* **2022**, *11*, 74–78. [CrossRef]
34. Li, H.; Liu, J.; Zhou, X. Intelligent map reader: A framework for topographic map understanding with deep learning and gazetteer. *IEEE Access* **2018**, *6*, 25363–25376. [CrossRef]
35. Ren, F.; Hou, W. Identification Method of Map Name Annotation Category for Machine Reading. *Geomat. Inf. Sci. Wuhan Univ.* **2020**, *45*, 273–280.
36. Girshick, R.; Donahue, J.; Darrell, T.; Berkeley, U.C.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal.* **2015**, *37*, 1904–1916. [CrossRef]

38. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
39. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
40. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
41. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Cham, Germany, 11–14 October 2016; pp. 21–37.
42. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Doll, A.R.P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
43. Guo, M.Q.; Bei, W.J.; Huang, Y.; Chen, Z.L.; Zhao, X.Z. Deep learning framework for geological symbol detection on geological maps. *Comput. Geosci.* **2021**, *157*, 104943. [CrossRef]
44. Qiu, Q.J.; Tan, Y.J.; Ma, K.; Tian, M.; Xie, Z.; Tao, L.F. Geological symbol recognition on geological map using convolutional recurrent neural network with augmented data. *Ore Geol. Rev.* **2023**, *153*, 105262. [CrossRef]
45. Schnürer, R.; Sieber, R.; Schmid-Lanter, J.; Öztireli, A.C.; Hurni, L. Detection of Pictorial Map Objects with Convolutional Neural Networks. *Cartogr. J.* **2021**, *58*, 50–68. [CrossRef]
46. Lehmann, R. 3 σ-rule for outlier detection from the viewpoint of geodetic adjustment. *J. Surv. Eng.* **2013**, *139*, 157–165. [CrossRef]
47. Lesperance, M.L.; Kalbfleisch, J.D. An Algorithm for Computing the Nonparametric MLE of a Mixing Distribution. *J. Am. Stat. Assoc.* **1992**, *87*, 120–126. [CrossRef]
48. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

An Integrated Framework for Landscape Indices' Calculation with Raster–Vector Integration and Its Application Based on QGIS

Yaqi Huang [1], Minrui Zheng [2,*], Tianle Li [1], Fei Xiao [1] and Xinqi Zheng [1,3]

[1] School of Information Engineering, China University of Geosciences (Beijing), Beijing 100083, China; huangyq@email.cugb.edu.cn (Y.H.); 3004200007@email.cugb.edu.cn (T.L.); xf@infomail.mnr.gov.cn (F.X.); zhengxq@cugb.edu.cn (X.Z.)
[2] School of Public Administration, Renmin University of China, Beijing 100872, China
[3] Beijing Fangshan Field Scientific Observation and Research Base of Comprehensive Exploration Technology, Ministry of Natural Resources, Beijing 100083, China
* Correspondence: minruizheng@ruc.edu.cn

Abstract: Landscape-index calculation tools play a pivotal role in ecosystem studies and urban-planning research, enabling objective assessments of landscape patterns' similarities and differences. However, the existing tools encounter limitations, such as the inability to visualize landscape indices spatially and the challenge of computing indices for both vector and raster data simultaneously. Based on the QGIS development platform, this study presents an innovative framework for landscape-index calculation that addresses these limitations. The framework seamlessly integrates both vector and raster data, comprising three main modules: data input, landscape-index calculation, and visualization. In the data-input module, the tool accommodates various data formats, including vector, raster, and tabular data. The landscape indices' calculation module allows users to select indices at patch, class, and landscape scales. Notably, the framework provides a comprehensive set of 165 indices for vector data and 20 for raster data, empowering users to selectively calculate landscape indices for vector or raster data to their specific needs and leverage the strengths of each data type. Moreover, the landscape-index visualization module enhances spatial visualization capabilities, meeting user demands for an insightful analysis. By addressing these challenges and offering enhanced functionalities, this framework aims to advance landscape indices' development and foster more comprehensive landscape analyses. And it presents a novel approach for landscape-index development.

Keywords: raster–vector integration; landscape indices' visualization; calculation tools; measure landscape patterns

1. Introduction

The landscape pattern encompasses the combination, distribution, and spatial relationships of landscape elements that vary in size and shape within a specific region. These patterns emerge from intricate interactions among a range of physical, biological, and social factors [1,2]. These patterns not only effectively reflect various ecological processes, such as species richness and ecosystem resilience to disturbances [3,4], but also represent a key focus in landscape ecology research, aiming to elucidate the interconnectedness between landscape patterns and ecological processes [5,6]. Consequently, the accurate assessment of landscape patterns and their relationships with ecological processes poses a significant challenge. In response, methodologies like the Patch Matrix Model (PMM) and Gradient Model (GM) have been proposed [1,7–11]. Landscape indices derived from the PMM model exhibit specific statistical properties. They quantitatively delineate the structure of the landscape by assessing its spatial configuration and composition. These indices are instrumental in landscape evaluation and planning processes [12].

To quantitatively assess landscape patterns, a multitude of methodologies have emerged in the academic literature. For instance, drawing inspiration from information theory, landscape indices like the Shannon diversity index and Simpson diversity index have been proposed to measure ecosystem and species diversity [13–16]. Moreover, concepts originating from fractal geometry in physics have been adapted for ecological analysis, employing methods such as area–perimeter analysis to measure the complexity of patch shapes [17,18]. Additionally, graph theory, borrowed from operations research, has found widespread application in ecological studies to explore landscape connectivity [19–21]. Drawing parallels with circuit theory, computations involving "resistance" or "current" have been employed to simulate ecological processes within heterogeneous landscapes, aiding in the identification of ecological corridors, bottlenecks, and barriers [22–24]. Despite the proliferation of landscape indices during this period, many of them exhibit redundancy. To address this concern, researchers have investigated correlations among landscape indices, utilizing the principal component analysis to derive a relatively independent set of indices reflecting landscape characteristics [25,26]. These metrics find widespread application in evaluating diverse landscapes or tracking changes in landscapes over time [27,28], such as monitoring ecological environmental changes [29]; studying species diversity [30]; and providing valuable information for urban expansion, farmland planning, and nature conservation [31–33].

Recognizing the importance of quantifying landscape patterns, it becomes necessary to acquire a suitable tool for quantifying various aspects of the landscape. Several software packages are available for computing landscape-pattern indices from either raster or vector data. For instance, Fragstats [34] offers computations from raster data, while VecLI [35] specializes in vector data calculation. Raster-based evaluation is more commonly favored due to its faster computation speed, particularly when raster resolution is adequate, in contrast to vector data processing. While conversion between raster and vector formats is feasible, it is generally discouraged [36,37] due to the potential introduction of errors into modeling and research relying on landscape indices. Furthermore, Conefor Sensinode 2.2 [38] is a notable software package for calculating landscape connectivity based on graph structure data, requiring node and connection data for computation. To aid in spatial data processing, GIS desktop software often integrates modules for computing landscape indices, such as the r.le [39] and r.li packages in GRASS GIS, Patch Analyst [40], v-Late [41] and Arc_LIND [42] plugins in ArcGIS, and the LecoS [43] plugin in QGIS. These plugins utilize the mature data-processing tools within GIS desktop software to preprocess input data before being processed by the plugins themselves. These plugins are capable of post-processing the output results generated by them. However, the condition to use these plugins is that you have already installed the corresponding GIS software on your computer and are familiar with its operation. This requirement facilitates easier usage for users familiar with the GIS software. Additionally, there are options such as Land_metrics DIY [44], a landscape index calculation function library written in C#, and landscapemetrics [45], another function library written in R, providing a broader array of landscape index options. But they necessitate higher programming proficiency from users. For a comparative overview of these tools, please refer to Table 1.

After enumerating the commonly utilized tools, we can outline the essential characteristics necessary for an effective landscape-index calculation program: (1) The program's existence as an independent application proves more efficient than those reliant on desktop GIS plugins. (2) It should be freely available for use, accessible to anyone who wishes to obtain and utilize it. (3) It should have the capability to process data in multiple formats, reducing errors in data conversion and ensuring data integrity. (4) The option to present index calculation results spatially should be considered. Many existing tools yield results in tabular form, hindering the analysis of spatial-distribution characteristics. If possible, we should transition from spatial data -> tables to spatial data -> spatial data to make the spatial distribution of indices more apparent. After a comprehensive analysis, the existing software for calculating landscape indices has the following issues: (1) Some of the software

is not openly accessible. (2) It processes and computes landscape indices for either raster or vector data types but not both. (3) It mostly presents landscape characteristics in tabular form, lacking spatial data representation of landscape index distributions. To address the aforementioned issues, this study implemented the following efforts:

- Based on the Quantum Geographic Information System (QGIS) [46] platform, this study conducted a secondary development to explore methods for integrating vector and raster data for landscape indices. Users can choose to calculate landscape indices using either vector or raster methods according to their needs.
- For vector data, 165 indices are available for selection, while for raster data, 20 indices are available.
- Three rendering methods are provided for visualizing landscape indices: unique value rendering, graduated rendering, and chart rendering.

Table 1. Differences of commonly used landscape-index calculation tools.

Name	Type	License	Input Data Format	Analysis Scale	Indices Type (Number)	Results Format
r.le	GRASS GIS plugin	FOSS (free and open-source software)	Raster data	Patch, class, landscape	Distance (16), attribute (6), size (6), shape (9), fractal (1), perimeter (6), diversity (4), texture (12), juxtaposition (2), edge (1)	Table
Fragstats	Standalone program	Freeware	Raster data	Patch, class, landscape	Area–edge (48), shape (87), core area (58), contrast (21), aggregation (74), diversity (9)	Table
Patch Analyst	ArcGIS plugin	Commercial	Vector data	Class, landscape	Area (4), density–size (10), edge (6), shape (10), diversity (2), core area (6)	Table
v-Late	ArcGIS plugin	Commercial	Vector data	Patch, class, landscape	Area–perimeter (8), form (9), edge (6), core area (9), nearest neighbor (1), proximity (3), diversity (7), subdivision (3)	Attribute table of shapefile, table
Conefor Sensinode 2.2	Standalone program	FOSS	Raster data and the focal species' movement-ability data	Landscape	Binary indices (6), probabilistic indices (3)	dbf
Land_metrics DIY	Library	FOSS	Vector data	Patch, class, landscape	Area–edge (12), core area (10), shape (2), aggregation (5), diversity (8)	Table
r.li	GRASS GIS plugin	FOSS	Raster data	Class, landscape	patch number (2), Area (4), shape (1), edge (2), patch attributes (2), diversity (6)	Table
LecoS	QGIS plugin	FOSS	Raster data	Class, landscape	Landscape statistics (11), patch statistics (20) area and edge (23)	Table
Arc_LIND	ArcGIS plugin	Commercial	Vector data	Patch, class, landscape	Shape (54) Core area (39). contrast (17), aggregation (53), diversity (9)	Table
landscapemetrics	Library	FOSS	Raster data	Patch, class, landscape	Area–edge (24), core area (28), shape (37), aggregation (30), diversity (14)	Output
VecLI	Standalone program	Freeware	Vector data	Patch, class, landscape	Area–edge (36), shape (54), core area (46), contrast (17), aggregation (55), diversity (9)	Table

2. Methods

2.1. Development Environment and System Architecture

QGIS is a freely available open-source desktop software for geographic information systems [47]. It has been an official project of the Open-Source Geospatial Foundation (OSGeo) since 2004. Compared to ArcGIS, which imposes user licensing restrictions on many of its functional modules, QGIS offers unrestricted accessibility to its functional modules. This characteristic makes QGIS more appealing to users, leading to a larger user base.

QGIS is primarily developed using the C++ programming language, supporting not only compilation and further development in C++, but also providing extension support for Python programming [48]. This allows for the execution of plugins developed with Python. Additionally, QGIS boasts a robust set of application programming interfaces (APIs) written in C++. Alongside the C++ API, QGIS also offers a Python API, enabling users to access the objects and methods of the C++ API through Python code. This Python function library of QGIS is commonly known as PyQGIS. All PyQGIS libraries are organized within a package called "qgis", which consists of four main sub-packages: qgis.core, qgis.gui, qgis.analysis, and qgis.server. In this study, the qgis.core and qgis.gui packages from QGIS version 3.28.12 are primarily utilized. The qgis.core package is employed to read, process, and visualize raster and vector data, and the qgis.gui package is used to design the interface for software tools. Furthermore, the PyQt5 package, which is a GUI extension specifically provided by Qt, a C++ Graphical User Interface application framework, for Python, is used in the program. PyQt5 is licensed under the GPL, ensuring no software copyright or revenue issues, thereby allowing developers to confidently utilize PyQt5 for software development. PyQt5 comprises 15 modules, with the fundamental 3 being QtCore, QtGui, and QtWidgets, which are used for handling and operating controls within various interfaces.

To achieve the goal, Vector and Raster Landscape Indices (VARLI) software (Developers: Yaqi Huang; Software available: https://drive.google.com/file/d/1v0DPBofGeV0tGDy61zI8E_G0Lo_TaBHm/view?usp=sharing (accessed on 3 July 2024); Programming language: Python 39; Integrated development environment: PyCharm; Program size: 200 MB) was designed through secondary development in QGIS. The class diagram of VARLI is shown in the Figure 1; it is structured around the core class "Landscape_indices_calculation". It is based on the data obtained from the "Data_input" class and the parameters input by the user through the "Parameter_input" class. The program then invokes the calculation functions corresponding to the indices from the "Vector_indices_function_library" class and "Raster_indices_function_library" class. Finally, it can output the calculated indices results and visualize them spatially.

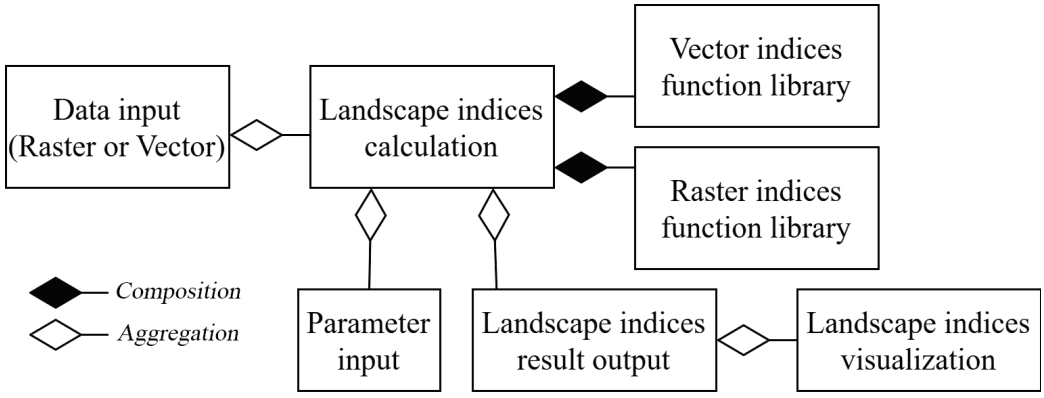


Figure 1. Class diagram of the VARLI. Black diamond (composition) represents the relationship between the whole and its parts, where the parts cannot exist independently of the whole; the black diamond points to the whole. White diamond (aggregation) represents the relationship between the whole and its parts, where the parts can exist independently of the whole; the white diamond points to the whole.

2.2. Raster–Vector Integration Framework

Spatial data are commonly categorized into two main structures: vector data and raster data. Vector data utilize points, lines, polygons, and their combinations from Euclidean geometry to represent geographical entities. This form of data organization enables precise definition of location, length, area, and other attributes of geographical entities. It is characterized by the implicit representation of attributes and clear delineation of positions. On the other hand, raster data organize the landscape into regular grids, with each grid cell assigned at least one corresponding attribute value to represent geographical entities. Each grid cell corresponds to one attribute in a band, representing a discrete numerical value in space. Raster data are characterized by the implicit representation of positions and clear delineation of attributes.

In PMM, landscape indices calculated from vector data tend to be more precise; however, the data structure is more complex, so computers require more time to process it. Conversely, landscape indices derived from raster data demonstrate scale effects, wherein the sizes of raster cells impact the resulting indices. Nonetheless, raster data structures are comparatively simpler, resulting in quicker computations under lower raster resolutions. For high-precision requirements, it is recommended to compute landscape indices using vector data. Conversely, when approximate results suffice, raster data can be employed. Consider two scenarios: if a user possesses vector data and requires precise results, direct computation of landscape indices in vector format is recommended. However, for expedient decision-making, converting vector data into raster format by adjusting raster cell sizes is advantageous. In another scenario, if the user has high-resolution raster data with slow processing speeds, converting them to vector format is a viable option. Nevertheless, at lower data resolutions, converting raster data to vector may introduce transformation errors, suggesting that using raster data for computing landscape indices remains a preferable choice.

Fragstats [34] encompasses a comprehensive range of landscape indices, several of which are widely utilized by scholars. Consulting the landscape indices provided in Fragstats, the landscape indices' summary provided by this tool is outlined in Table 2, featuring acronyms for each index. Detailed descriptions of these indices can be found in Appendices A and B. At the vector data level, 165 landscape indices are calculated, covering dimensions such as area–edge, core area, contrast, shape, aggregation, and diversity across patch, class, and landscape scales (as indicated in the green and blue sections of Table 2). For raster data, approximately 238 landscape indices are computed across the same dimensions and scales (as depicted in the green, blue, and orange sections of Table 2). Certain landscape indices require adjustments based on data type. For instance, indices such as SHAPE, FRAC, and LSI, highlighted in the blue section of Table 2, require correction for perimeter–area ratio discrepancies. For raster data, a square standard with a correction parameter of 0.25 is adopted to rectify size discrepancies in perimeter–area ratios. Conversely, for vector data, a circular standard is employed for this purpose. Formulas and explanations for these calculations are provided in Appendix B.

The integrated vector–raster framework devised is depicted in Figure 2. Users have the flexibility to input either vector or raster data for landscape indices' calculation. Before computing the landscape indices, the tool conducts an additional check: is there a need to convert the data format? Subsequently, the tool employs functions corresponding to the data type in the landscape-index calculation function library to compute the landscape indices and obtain the calculation results. For vector data, landscape indices are computed by invoking the API for geometric features of vector elements. The code for each index is delineated within the Vector_indices_function_library depicted in Figure 1. Conversely, raster data require the use of matrix operations to calculate landscape indices. The code for each index within this context is encapsulated within the Raster_indices_function_library, as illustrated in Figure 1. By employing different methods for calculating landscape indices for vector and raster data, not only can the problem of data loss during vector–raster conversion be avoided, thereby reducing errors in landscape quantification, but it also

provides a choice: for large datasets of vector data, converting them into raster format for calculation is a viable option. This is particularly advantageous when the raster-cell size is significantly smaller than the overall landscape extent, making the disparities between landscape indices computed from vector and raster data negligible. Hence, opting for the computationally efficient raster data format for calculating landscape indices emerges as the preferred choice. Moreover, when computing edge-related landscape indices for raster data, converting them into vector format can effectively address concerns regarding biased edge values.

Table 2. Summary table of landscape indices. (The expanded form of these indices can be found in Appendices A and B [34]), Green cells indicate landscape indices calculated at both the vector and raster data levels, orange cells indicate landscape indices calculated at the raster data level, and blue cells indicate that the calculation method of this landscape indices differs between the vector and raster data levels. And statistics include six statistical indicators: MN, AM, MD, RA, SD, and CV.

	Area–Edge	Shape	Core Area	Contrast	Aggregation		
Patch	AREA	PARA	CORE	ECON	ENN		
	PERIM	SHAPE	NCORE		PROX		
	GYRATE	FRAC	CAI		SIMI		
		CIRCLE					
		CONTIG					
	Area–Edge	**Shape**	**Core Area**	**Contrast**	**Aggregation**	**Statistics**	
Class	CA/TA	PAFRAC	TCA	CWED	CONNECT	AREA	
	PLAND		CPLAND	TECI	NP	GYRATE	
	LPI		NDCA		PD	PARA	
	TE		DCAD		DIVISION	SHAPE	
	ED				SPLIT	FRAC	
					MESH	CIRCLE	
					IJI	CONTIG	
					PLADJ	CORE	
					AI	DCORE	
					CLUMPY	CAI	
					LSI	ECON	
					NLSI	ENN	
					COHESION	PROX	
						SIMI	
	Area–Edge	**Shape**	**Core Area**	**Contrast**	**Aggregation**	**Diversity**	**Statistics**
Landscape	CA/TA	PAFRAC	TCA	CWED	CONNECT	PR	AREA
	LPI		NDCA	TECI	NP	PRD	GYRATE
	TE		DCAD		PD	RPR	PARA
	ED				DIVISION	SHDI	SHAPE
					SPLIT	SIDI	FRAC
					MESH	MSIDI	CIRCLE
					CONTAG	SHEI	CONTIG
					IJI	SIEI	CORE
					PLADJ	MSIEI	DCORE
					AI		CAI
					LSI		ECON
					COHESION		ENN
							PROX
							SIMI

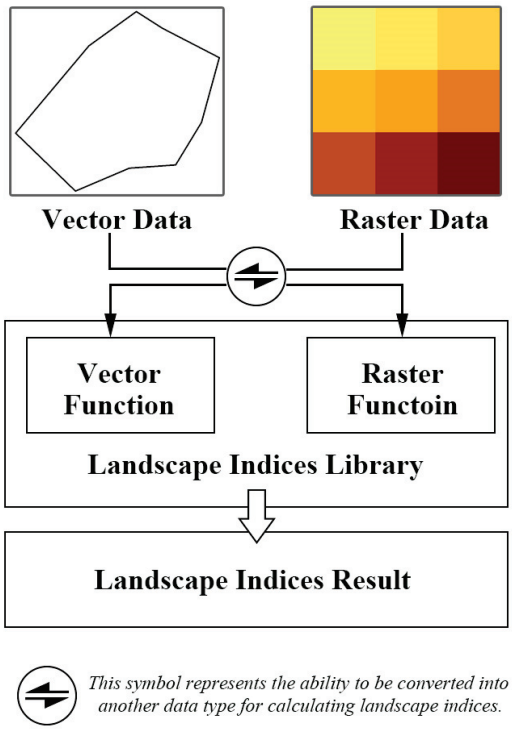


Figure 2. Integrated vector–raster landscape-index calculation method.

2.3. Landscape-Index Visualization

Visualizing the results of landscape indices' calculations plays a crucial role in enhancing the comprehension of spatial information within geographic landscape patterns. This visualization enables the observation of spatial distribution and patterns in the data, facilitating decision-makers in quickly grasping the situation and making more informed decisions. In VARLI, the calculation results of landscape indices for vector data can be added to its attribute table, enabling feature-based visualization. Additionally, for raster data, the calculation results of landscape indices are stored as bands, allowing for spatial visualization of the indices in the form of a raster matrix. The visualization methods (Figure 3) primarily consist of three types: unique value rendering, graduated rendering, and chart rendering.

2.3.1. Unique Value Rendering

The unique value-rendering method displays data by assigning unique symbols, colors, and styles to different categories. This method is commonly used for discrete data. Class-level landscape indices typically represent statistical information about patches of different landscape categories. Specifically, for patches belonging to the same category but different locations, the same class-level landscape indices usually have the same value. Therefore, this rendering method is generally employed to display class-level landscape indices, helping to distinguish and visualize the spatial differences in index calculation results for different landscape categories, which facilitate further analysis.

2.3.2. Graduated Rendering

Graduated rendering is used to classify data into different categories based on the distribution of their attribute values. Common classification methods include equal interval, natural breaks, geometric intervals, and standard deviation. After classification, each category is assigned a distinct symbol or color to clearly display the differences and distribution of the data. This rendering method is typically used for continuous data. Patch-level landscape indices, which are often used to represent the landscape-attribute information of different patches within a landscape, are numerous and constitute continuous

data. Therefore, this rendering method is generally utilized to express spatial distribution of patch-level landscape indices. However, when necessary, it can also be used to display the spatial distribution of class-level landscape indices. Different patches have different landscape indices based on their shapes and spatial characteristics, and visualizing the landscape-index calculation results of each patch spatially facilitates further analysis of the spatial distribution characteristics of the landscape pattern in the area.

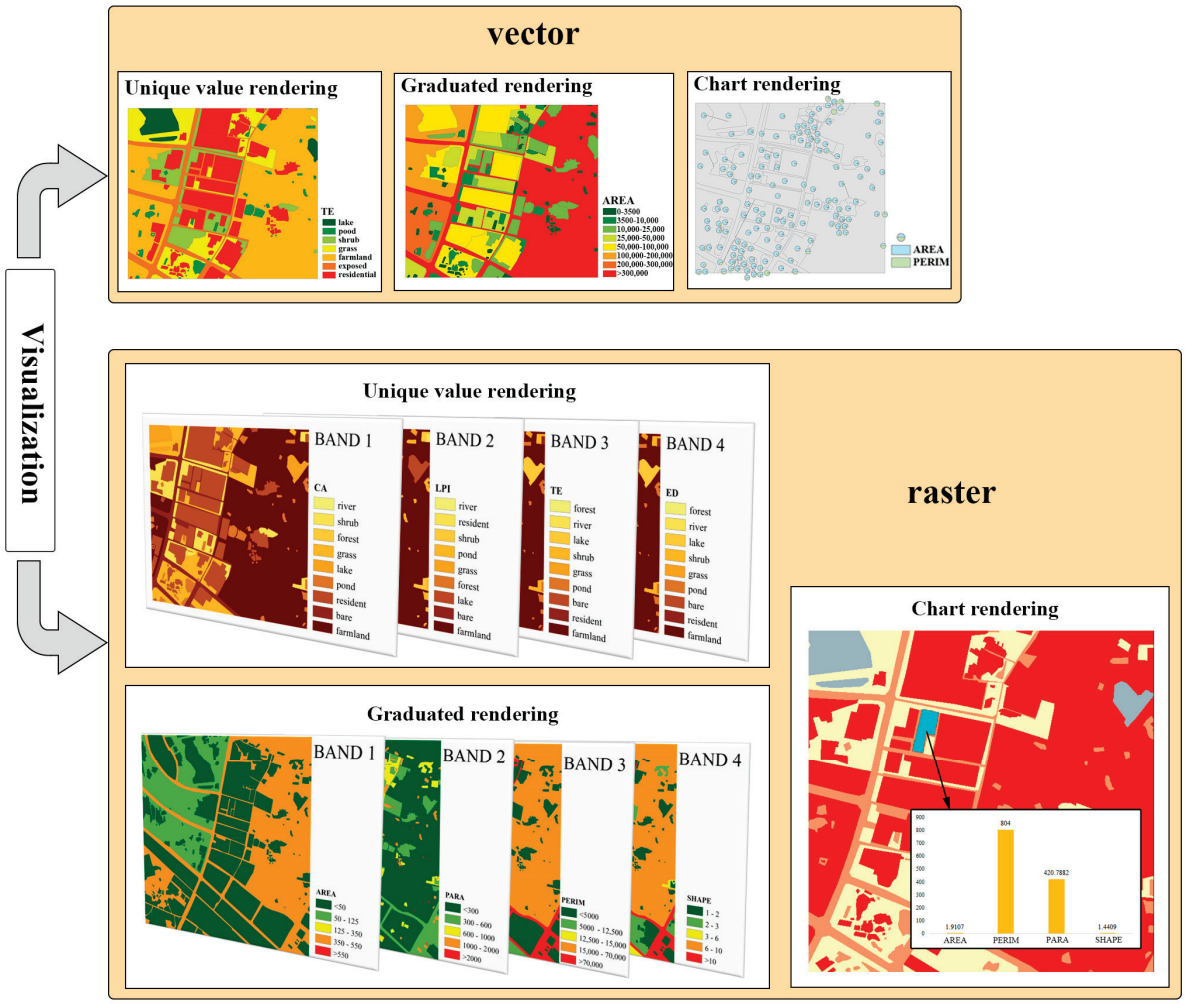


Figure 3. Visualization methods.

2.3.3. Chart Rendering

Chart rendering is utilized to present data on a map in the form of charts, including bar charts, line charts, pie charts, scatter plots, etc. This method enables the display of different landscape indices of the same patch at the same level (such as patch scale, class scale, or landscape scale) in the chart, or the depiction of landscape indices of the same patch at different levels. For instance, comparing the patch-level and class-level landscape indices of a patch through chart rendering can intuitively reflect the spatial information in the landscape pattern. This approach facilitates the visualization and analysis of landscape indices' data, aiding in the interpretation of spatial patterns and relationships within the landscape.

3. Program Description

3.1. Data Input Module

In the interface, buttons were added to enable the loading of vector and raster data. The current version of VARLI allows for loading vector data in SHP format and raster data

in GeoTIFF format. In future versions, the supported vector data inputs' data formats will include shapefile (SHP), Keyhole Markup Language (KML), GeoJSON, and file geodatabase formats stored in databases. For raster data, supported formats such as GeoTIFF, ASCII Grid, and IMG were used. This section implements functionality by designing functions triggered by button clicks. A mapCanvas is provided in the main interface for displaying vector and raster spatial data. Further functionalities will be developed and implemented in subsequent versions of VARLI.

3.2. Landscape Indices' Calculation Module

During the landscape indices' calculation process, the method of raster–vector integration is integrated into VARLI. Users need to input the following parameters: (a) the data layers for which landscape indices need to be calculated; (b) determination of whether the data layer to be calculated is vector or raster data; (c) determination of whether the user needs to convert vector data to raster data or raster data to vector data; (d) the parameters required for calculating landscape indices; and (e) when calculating landscape indices at the class scale, selection of the corresponding classification field to determine the landscape type to which each patch belongs. VARLI requires these five parameters for landscape indices' calculation. This model offers users a choice: they can obtain landscape-index calculation results either in vector data format or raster data format. The calculations for various functions are contained within a calculation class, which contains functions for calculating each landscape indices. Currently, VARLI can provide the calculation of all landscape indices for vector data and calculation of a few landscape indices for raster data. Additional landscape-index calculation functionalities will be gradually improved in subsequent versions of VARLI.

3.3. Landscape Indices' Visualization Module

In this module, VARLI enables the display of calculation results of patch-scale, class-scale, and landscape-scale landscape indices in tabular form. Additionally, it offers functionality for exporting data to Excel. After obtaining the calculation results, users can select from various visualization methods to render the indices' results visually within VARLI. Currently, VARLI offers three rendering methods: single-symbol rendering, unique-value rendering, and graduated rendering. More diverse visualization methods will be provided in future versions for enhanced functionality.

4. Case Study

Based on 2020 Google Earth remote-sensing data, this study selected Yaopu Town in Chuzhou City, Anhui Province, China, as a case study. The purpose of this part is to explain the effectiveness of the VARLI. Through visually interpreting and vectorizing the remote-sensing data of Yaopu, obtain the land use data for it in 2020 (Figure 4b). Yaopu is predominantly agricultural, with a significant amount of farmland and rich water resources. The land use in Yaopu is diverse and can be categorized into nine types: grassland, farmland, shrubland, rivers, lakes, residential areas, ponds, bare land, and forests.

The landscape of Yaopu was measured using three indices: SPLIT (splitting index), SHDI (Shannon's diversity index), and PLAND (percentage of landscape). SPLIT is used to assess the degree of landscape fragmentation; a higher SPLIT value indicates more severe fragmentation of patches within the landscape, while a lower SPLIT value suggests that the patches are more aggregated and continuous. SHDI measures the quantity and distribution of different landscape types, reflecting the landscape's heterogeneity. PLAND indicates the dominant land use type within the landscape.

The visualization of SPLIT indices for different land use types is rendered using unique values (Figure 4c). This allows for a direct observation of the fragmentation levels of different land use types. To capture local phenomena effectively, a grid with 200 m sides was created to cover the study area. The SPLIT index measured fragmentation within each grid cell and was visualized using graduated rendering (Figure 4d). Consistent with

Figure 4c, the central and western regions of the study area exhibit higher landscape fragmentation. Based on the locations of the seven communities and villages within the study area, the area was divided into different administrative regions using Thiessen polygons. Then, the SHDI indices for these regions were calculated and visualized using graduated rendering (Figure 4e). The results suggest that the SHDI values for Yaopu Community and Meipu Community are relatively higher compared to the results of other regions. This indicates a greater diversity and more even distribution of landscape types, which may imply higher landscape heterogeneity in these communities. Additionally, the PLAND metric was used to measure the proportion of different land use types across various regions (Figure 4f). It can be observed that farmland predominates the overall study area, with higher forest cover in the western area. Grassland is more prevalent in Meipu Community and Yaopu Community compared to other areas, and Dongchen has a significant proportion of lake coverage.

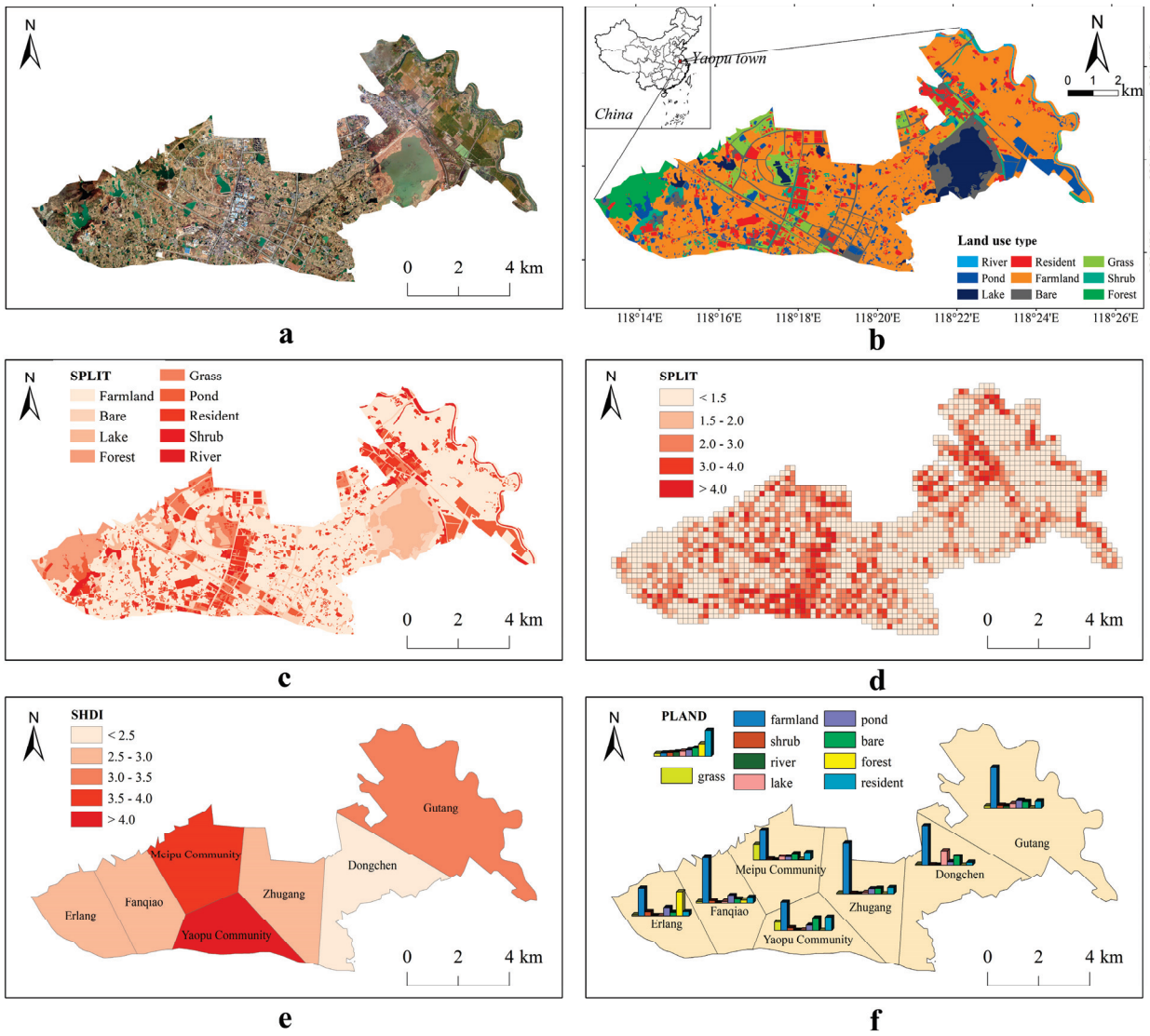


Figure 4. Spatial distribution visualization results of landscape indices based on land use data for Yaopu Town. (**a**) Google Earth remote-sensing imagery of Yaopu Town in 2020. (**b**) Land use data of Yaopu Town in 2020. (**c**) SPLIT results for different land use types (unique value rendering). (**d**) Grid-scale SPLIT results (graduated rendering). (**e**) Regional-scale SHDI results (graduated rendering). (**f**) Regional-scale PLAND results for different land use types (chart rendering).

The choice of visualization-rendering methods may vary depending on different objectives. But the method of spatial visualization for landscape indices demonstrates the spatial distribution of landscape indices. Since the results are written into the attribute tables of vector features and the bands of raster features, this provides a foundation for further spatial analysis, such as Moran's I index analysis and geographically weighted regression.

5. Discussion

This paper proposes a framework for calculating landscape indices by integrating vector and raster data. It includes methods for visualizing these landscape indices. Using Yaopu Town as a case study, the spatial distribution of landscape indices is analyzed from the perspectives of landscape type, grid, and region. It demonstrates that visualizing landscape indices is beneficial for analyzing their spatial distribution characteristics.

This paper calculates landscape indices in different dimensions by referencing the formulas provided by Fragstats. Some formulas of landscape indices based on vector data are adjusted according to vector standards in VARLI. Currently, VARLI offers a rich array of landscape indices for vector data, and it will continue to expand the landscape indices of raster data in future work.

Based on the vector data of Yaopu Town, the data were converted into raster data with a 1 m pixel size. Then, the landscape-index results were obtained using VARLI, VecLI, and Fragstats software, respectively. The calculation results from VARLI were compared with those from VecLI and Fragstats, using three statistical measures: Standard Deviation of Absolute Error, Pearson Correlation, and *t*-test (Table 3). VecLI and Fragstats provide a comprehensive range of landscape indices for both vector and raster data. Comparing VARLI's landscape indices with those from VecLI and Fragstats can verify the accuracy of VARLI's results.

Table 3. Comparison of landscape-index results between VARLI, VECLI, and Fragstats.

Dimension	Acronym	VECLI			Fragstats		
		Standard Deviation of Absolute Error	Pearson Correlation	*T*-Test (*p*-Value)	Standard Deviation of Absolute Error	Pearson Correlation	*T*-Test (*p*-Value)
Area	AREA	0.001	1.000	0.196	0.004	1.000	0.718
	CA	0.000	1.000	0.460	3.464	1.000	0.211
	PLAND	0.000	1.000	0.558	0.046	1.000	1.000
	CPLAND	10.665	0.066	0.233	0.132	1.000	0.174
	LPI	3.650	1.000	0.048	3.678	0.998	0.045
Edge	PERIM	3.201	1.000	0.151	3.787	1.000	0.318
	GYRATE	-	-	-	3.708	0.999	0.000
	TE	50,642.248	0.991	0.012	40,721.756	1.000	0.015
	ED	6.328	0.991	0.012	5.137	1.000	0.015
Shape	PARA	0.000	1.000	0.181	0.015	1.000	0.322
	SHAPE	0.003	1.000	0.000	0.059	0.998	0.000
	FRAC	0.015	0.993	0.000	0.253	−0.620	0.000
	CIRCLE	0.169	0.590	0.000	0.136	0.885	0.000
	CONTIG	-	-	-	0.000	1.000	0.004
	PAFRAC	-	-	-	0.071	0.905	0.080
Core area	CORE	25.826	0.362	0.027	0.124	1.000	0.196
	NCORE	1.630	0.231	0.000	3.091	0.972	0.054
	CAI	4.997	0.944	0.000	0.125	1.000	0.028
	TCA	759.604	0.721	0.328	10.356	1.000	0.232
	NDCA	238.703	0.963	0.085	241.265	0.741	0.189
	DCAD	-	-	-	3.012	0.741	0.189

Table 3. *Cont.*

Dimension	Acronym	VECLI			Fragstats		
		Standard Deviation of Absolute Error	Pearson Correlation	*T*-Test (*p*-Value)	Standard Deviation of Absolute Error	Pearson Correlation	*T*-Test (*p*-Value)
Contrast	ECON	14.720	0.109	0.001	10.382	0.487	0.000
	CWED	-	-	-	2.253	1.000	0.013
	TECI	-	-	-	0.590	0.996	0.874
Aggregation	ENN	-	-	-	0.631	1.000	0.317
	PROX	-	-	-	232.492	0.939	0.144
	SIMI	-	-	-	1293.405	0.928	0.000
	CONNECT	0.025	1.000	0.704	10.254	−0.313	0.468
	NP	0.667	1.000	0.195	24.188	0.997	0.910
	PD	0.008	1.000	0.195	0.300	0.997	0.939
	DIVISION	0.000	1.000	0.861	0.000	1.000	0.273
	SPLIT	1.286	1.000	0.350	506.067	1.000	0.343
	MESH	958,794.251	1.000	0.274	2.991	1.000	0.310
	IJI	3.795	0.939	0.118	2.222	0.990	0.014
	LSI	1.050	0.992	0.011	7.180	0.763	0.001
Diversity	PR	149.083	−0.545	0.172	0.000	1.000	0.153
	PRD	17.725	0.311	0.182	0.002	1.000	0.202
	RPR	-	-	-	0.000	1.000	0.000
	SHDI	0.383	−0.152	0.043	0.004	1.000	0.085
	SIDI	0.137	0.241	0.191	0.001	1.000	0.076
	MSIDI	0.709	0.400	0.185	0.004	1.000	0.112
	SHEI	0.035	0.925	0.246	0.002	1.000	0.084
	SIEI	0.102	0.511	0.241	0.001	1.000	0.077
	MSIEI	0.055	0.910	0.319	0.002	1.000	0.117
Area	AREA_MN	0.012	1.000	0.316	18.174	0.908	0.793
	AREA_AM	117.321	0.811	0.025	31.441	0.986	0.337
	AREA_MD	0.330	0.999	0.154	43.436	0.587	0.414
	AREA_RA	0.005	1.000	0.245	17.449	0.998	0.481
	AREA_SD	0.006	1.000	0.301	16.892	0.963	0.627
	AREA_CV	0.465	1.000	0.254	41.441	0.940	0.310
Shape	SHAPE_MN	0.082	1.000	0.000	0.304	0.821	0.048
	SHAPE_AM	287.449	0.357	0.008	283.234	0.993	0.008
	SHAPE_MD	0.102	0.998	0.000	0.298	0.866	0.086
	SHAPE_RA	0.739	1.000	0.032	1.877	0.982	0.105
	SHAPE_SD	0.120	1.000	0.017	0.301	0.961	0.214
	SHAPE_CV	0.176	1.000	0.820	9.983	0.984	0.294
	FRAC_MN	0.004	0.999	0.000	0.038	0.809	0.003
	FRAC_AM	267.500	−0.800	0.068	267.241	0.002	0.068
	FRAC_MD	0.005	0.998	0.000	0.010	0.992	0.000
	FRAC_RA	0.215	0.678	0.326	0.075	0.922	0.168
	FRAC_SD	0.003	0.995	0.426	0.066	0.851	0.278
	FRAC_CV	0.274	0.994	0.866	4.631	0.870	0.306
	PARA_MN	0.028	0.878	0.321	6411.384	0.559	0.051
	PARA_AM	130.722	0.626	0.000	33.194	0.997	0.000
	PARA_MD	0.000	1.000	0.055	7621.365	−0.345	0.234
	PARA_RA	13.436	0.083	0.344	17,992.356	0.384	0.006
	PARA_SD	0.540	0.185	0.341	6741.836	0.731	0.016
	PARA_CV	248.362	0.235	0.334	110.976	−0.311	0.447
	CIRCLE_MN	-	-	-	0.056	0.853	0.028
	CIRCLE_AM	-	-	-	1266.739	0.994	0.000
	CIRCLE_MD	-	-	-	0.028	0.969	0.036
	CIRCLE_RA	-	-	-	0.263	0.182	0.553
	CIRCLE_SD	-	-	-	0.075	0.382	0.060
	CIRCLE_CV	-	-	-	13.377	0.414	0.039

Table 3. *Cont.*

Dimension	Acronym	VECLI			Fragstats		
		Standard Deviation of Absolute Error	Pearson Correlation	*T*-Test (*p*-Value)	Standard Deviation of Absolute Error	Pearson Correlation	*T*-Test (*p*-Value)
Core area	CORE_MN	11.864	0.892	0.326	13.600	0.931	0.724
	CORE_AM	138.619	0.068	0.068	13.360	0.995	0.326
	CORE_MD	1.226	0.886	0.320	35.005	0.750	0.374
	CORE_RA	221.376	0.391	0.329	10.792	0.999	0.446
	CORE_SD	38.091	0.629	0.332	12.201	0.970	0.541
	CORE_CV	34.753	0.986	0.531	43.690	0.974	0.508
	DCORE_MN	0.560	0.768	0.006	6.435	0.475	0.127
	DCORE_AM	7.478	0.095	0.058	134.171	0.375	0.058
	DCORE_MD	0.471	0.919	0.081	1.020	−0.128	0.013
	DCORE_RA	13.392	0.647	0.228	227.703	0.323	0.076
	DCORE_SD	1.836	0.732	0.268	27.473	0.239	0.053
	DCORE_CV	53.334	0.034	0.256	100.369	0.811	0.000
	CAI_MN	13.455	0.977	0.006	16.544	0.880	0.020
	CAI_AM	138.694	−0.300	0.077	105.516	0.819	0.190
	CAI_MD	17.606	0.972	0.076	19.329	0.733	0.169
	CAI_RA	11.980	0.992	0.000	15.459	0.828	0.000
	CAI_SD	6.060	0.998	0.000	5.564	0.903	0.000
	CAI_CV	7.823	0.995	0.119	25.669	0.915	0.228
Contrast	ECON_MN	18.494	−0.222	0.029	1.205	0.984	0.759
	ECON_AM	-	-	-	0.541	0.998	0.961
	ECON_MD	20.720	−0.472	0.924	0.841	0.994	0.052
	ECON_RA	136.768	0.121	0.008	7.833	0.955	0.136
	ECON_SD	34.719	0.017	0.002	0.746	0.976	0.261
	ECON_CV	47.347	0.267	0.001	1.445	0.981	0.428

From Table 3, it is clear that VARLI exhibits a higher correlation and shows no statistically significant differences compared to VecLI and Fragstats in terms of indices related to the area dimension. In the edge dimension indices, VARLI shows a higher correlation with VecLI and Fragstats, and some indices display significant differences. This may be attributed to VARLI's method of calculating landscape edges, where it counts the boundary lines between adjacent patches only once. For the shape-dimension indices, it can be observed that there is a high correlation, yet significant differences exist. These differences can be attributed to the variations in the parameters used within the equations (Appendix B). The core area and contrast-dimension indices demonstrate a high correlation with those obtained from Fragstats. The phenomenon of significant differences in some indices could be attributed to the disparities between vector and raster data. Regarding the aggregation and diversity dimensions, it is clear that the results of VARLI are consistent with Fragstats, and it can demonstrate VARLI's accuracy.

In general, VARLI offers the following advantages:

- It provides rich landscape indices based on vector data and some indices based on raster data, addressing the limitation of existing software that only calculates landscape indices for a single data type.
- It includes three visualization methods for displaying landscape metric results.
- It is open-access and designed for ease of use.

On the other hand, VARLI has some limitations:

- It exhibits lower computational efficiency when processing large volumes of land use data. Thus, the algorithm still needs further optimization in the future.
- Currently, there are limitations for VARLI in displaying the landscape indices derived from raster data, primarily including those reported by Fragstats. In subsequent work,

it would be beneficial to focus on incorporating additional indices that are useful for ecological assessment into VARLI.

6. Conclusions and Further Work

We propose an integrated framework that generates landscape indices based on vector and raster data. This framework includes functionality for visualizing indices, allowing for the spatial visualization of indices, which facilitates the observation of their spatial distribution and further analysis. A comparison of the landscape indices calculated by VARLI with those obtained from VecLI and Fragstats shows that the results from VARLI are partially consistent with those from other open-source software, while also exhibiting some differences. To advance landscape ecology further, we recommend the implementation of the following functionalities in the future:

- Developing a web-based application or tool for calculating landscape indices: Currently, landscape indices are mostly calculated using locally installed software, which consumes computer memory. Users need to download and install the software when they need to use it. Subsequent work should further explore the possibility of calculating landscape indices on the web and analyzing and computing landscape indices in an online environment. This facilitates the sharing of index calculation methods, allowing for adjustments and improvements based on calculated results and user feedback. More importantly, it can achieve cross-platform compatibility for landscape-index calculations.
- The functionality of performing addition, subtraction, multiplication, and division operations among landscape indices can be enhanced, and if possible, relevant functions could also be introduced. While organizing the formulas for calculating landscape indices, it was discovered that most indices are composed of a few basic elements, such as FRAC and SHAPE, as detailed in Appendix B. Both indices are derived from the perimeter and area of patch, differing only in their respective functions. Since many indices are constructed from these fundamental elements, performing these operations can enhance user understanding of the indices. For instance, the indices PD and AREA_MN are highly correlated, and selecting both could lead to data redundancy. Adding this functionality will make users more considerate when making their selections. What is more, the ability to recombine these basic elements to create new indices could further unlock the potential value of landscape indices.
- Additional geospatial data mining tools can be incorporated. Subsequently, the system can be enhanced by integrating machine-learning algorithms such as logistic regression analysis, principal component analysis, support vector machines, neural networks, and random forest regression. I believe that integrating landscape index calculations with other geospatial data mining tools will provide model support for the in-depth exploration of the information contained within landscape indices and the analysis of landscape patterns.

Author Contributions: Conceptualization, Yaqi Huang, Minrui Zheng and Xinqi Zheng; data curation, Yaqi Huang; formal analysis, Fei Xiao; funding acquisition, Minrui Zheng and Xinqi Zheng; investigation, Xinqi Zheng; methodology, Yaqi Huang, Minrui Zheng, Tianle Li, Fei Xiao and Xinqi Zheng; software, Yaqi Huang and Tianle Li; supervision, Tianle Li; writing—original draft, Yaqi Huang; writing—review and editing, Minrui Zheng, Tianle Li, Fei Xiao and Xinqi Zheng. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 42201471; the Beijing Social Science Foundation, grant number 2022YJC264; the Key Program of National Natural Science Foundation of China, grant number 72033005; and the Third Xinjiang Scientific Expedition of the Key Research and Development Program by Ministry of Science and Technology of the People's Republic of China, grant number 2022xjkk1104.

Data Availability Statement: Data are available upon request to the authors.

Acknowledgments: The authors express gratitude to OpenAI for providing ChatGPT 3.5 to assist in writing the program, enabling the implementation of some functionalities within the program.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

List of full names of landscape indices calculated in the tool. (Referenced from Fragstats, with calculation formulas and detailed explanations provided)

Metric Number	Level	Acronym	Name
1	P	AREA	Patch area
2	P	PERIM	Patch perimeter
3	P	GYRATE	Radius of gyration
4	P	PARA	Perimeter–area ratio
5	P	SHAPE	Shape index
6	P	FRAC	Fractal dimension index
7	P	CIRCLE	Related circumscribing circle
8	P	CONTIG	Contiguity index
9	P	CORE	Core area
10	P	NCORE	Number of core areas
11	P	CAI	Core area index
12	P	ECON	Edge contrast index
13	P	ENN	Euclidean nearest-neighbor distance
14	P	PROX	Proximity index
15	P	SIMI	Similarity index
16	C	CA/TA	Total area (class)
17	C	PLAND	Percentage of landscape
18	C	CPLAND	Core area percentage of landscape
19	C	CLUMPY	Clumpiness index
20	C	NLSI	Normalized landscape shape index
21	L	CA/TA	Total area (landscape)
22	L	CONTAG	Contagion
23	L	PR	Patch richness
24	L	PRD	Patch richness density
25	L	RPR	Relative patch richness
26	L	SHDI	Shannon’s diversity index
27	L	SIDI	Simpson’s diversity index
28	L	MSIDI	Modified Simpson’s diversity index
29	L	SHEI	Shannon’s evenness index
30	L	SIEI	Simpson’s evenness index
31	L	MSIEI	Modified Simpson’s evenness index
32	C,L	LPI	Largest patch index
33	C,L	TE	Total edge
34	C,L	ED	Edge density
35	C,L	PAFRAC	Perimeter–area fractal dimension
36	C,L	TCA	Total core area
37	C,L	NDCA	Number of disjunct core areas
38	C,L	DCAD	Disjunct core area density
39	C,L	CWED	Contrast-weighted edge density
40	C,L	TECI	Total edge contrast index
41	C,L	CONNECT	Connectance index
42	C,L	NP	Number of patches
43	C,L	PD	Patch density
44	C,L	DIVISION	Landscape division index
45	C,L	SPLIT	Splitting index
46	C,L	MESH	Effective mesh size
47	C,L	IJI	Interspersion juxtaposition index

Metric Number	Level	Acronym	Name
48	C,L	PLADJ	Proportion of like adjacencies
49	C,L	AI	Aggregation index
50	C,L	LSI	Landscape shape index
51	C,L	COHESION	Patch cohesion index
52	C,L	AREA_MN	The mean of patch area
53	C,L	AREA_AM	The area-weighted mean of patch area
54	C,L	AREA_MD	The median of patch area
55	C,L	AREA_RA	The range of variation in patch area
56	C,L	AREA_SD	The standard deviation of patch area
57	C,L	AREA_CV	The coefficient of variance of patch area
58	C,L	GYRATE_MN	The mean of radius of gyration
59	C,L	GYRATE_AM	The area-weighted mean of radius of gyration
60	C,L	GYRATE_MD	The median of radius of gyration
61	C,L	GYRATE_RA	The range of variation in radius of gyration
62	C,L	GYRATE_SD	The standard deviation of radius of gyration
63	C,L	GYRATE_CV	The coefficient of variance of radius of gyration
64	C,L	SHAPE_MN	The mean of shape index
65	C,L	SHAPE_AM	The area-weighted mean of shape index
66	C,L	SHAPE_MD	The median of shape index
67	C,L	SHAPE_RA	The range of variation in shape index
68	C,L	SHAPE_SD	The standard deviation of shape index
69	C,L	SHAPE_CV	The coefficient of variance of shape index
70	C,L	FRAC_MN	The mean of fractal dimension index
71	C,L	FRAC_AM	The area-weighted mean of fractal dimension index
72	C,L	FRAC_MD	The median of fractal dimension index
73	C,L	FRAC_RA	The range of variation in fractal dimension index
74	C,L	FRAC_SD	The standard deviation of fractal dimension index
75	C,L	FRAC_CV	The coefficient of variance of fractal dimension index
76	C,L	PARA_MN	The mean of perimeter–area ratio
77	C,L	PARA_AM	The area-weighted mean of perimeter–area ratio
78	C,L	PARA_MD	The median of perimeter–area ratio
79	C,L	PARA_RA	The range of variation in perimeter–area ratio
80	C,L	PARA_SD	The standard deviation of perimeter–area ratio
81	C,L	PARA_CV	The coefficient of variance in perimeter–area ratio
82	C,L	CIRCLE_MN	The mean of related circumscribing circle
83	C,L	CIRCLE_AM	The area-weighted mean of related circumscribing circle
84	C,L	CIRCLE_MD	The median of related circumscribing circle
85	C,L	CIRCLE_RA	The range of variation in related circumscribing circle
86	C,L	CIRCLE_SD	The standard deviation of related circumscribing circle
87	C,L	CIRCLE_CV	The coefficient of variance of related circumscribing circle
88	C,L	CONTIG_MN	The mean of contiguity index
89	C,L	CONTIG_AM	The area-weighted mean of contiguity index
90	C,L	CONTIG_MD	The median of contiguity index
91	C,L	CONTIG_RA	The range of variation in contiguity index
92	C,L	CONTIG_SD	The standard deviation of contiguity index
93	C,L	CONTIG_CV	The coefficient of variance of contiguity index
94	C,L	CORE_MN	The mean of core area
95	C,L	CORE_AM	The area-weighted mean of core area
96	C,L	CORE_MD	The median of core area
97	C,L	CORE_RA	The range of variation in core area
98	C,L	CORE_SD	The standard deviation of core area
99	C,L	CORE_CV	The coefficient of variance of core area
100	C,L	DCORE_MN	The mean of disjunct core area
101	C,L	DCORE_AM	The area-weighted mean of disjunct core area
102	C,L	DCORE_MD	The median of disjunct core area
103	C,L	DCORE_RA	The range of variation in disjunct core area

Metric Number	Level	Acronym	Name
104	C,L	DCORE_SD	The standard deviation of disjunct core area
105	C,L	DCORE_CV	The coefficient of variance of disjunct core area
106	C,L	CAI_MN	The mean of core area index
107	C,L	CAI_AM	The area-weighted mean of core area index
108	C,L	CAI_MD	The median of core area index
109	C,L	CAI_RA	The range of variation in core area index
110	C,L	CAI_SD	The standard deviation of core area index
111	C,L	CAI_CV	The coefficient of variance of core area index
112	C,L	PROX_MN	The mean of proximity index
113	C,L	PROX_AM	The area-weighted mean of proximity index
114	C,L	PROX_MD	The median of proximity index
115	C,L	PROX_RA	The range of variation in proximity index
116	C,L	PROX_SD	The standard deviation of proximity index
117	C,L	PROX_CV	The coefficient of variance of proximity index
118	C,L	SIMI_MN	The mean of similarity index
119	C,L	SIMI_AM	The area-weighted mean of similarity index
120	C,L	SIMI_MD	The median of similarity index
121	C,L	SIMI_RA	The range of variation in similarity index
122	C,L	SIMI_SD	The standard deviation of similarity index
123	C,L	SIMI_CV	The coefficient of variance of similarity index
124	C,L	ENN_MN	The mean of Euclidean nearest-neighbor distance
125	C,L	ENN_AM	The area-weighted mean of Euclidean nearest-neighbor distance
126	C,L	ENN_MD	The median of Euclidean nearest-neighbor distance
127	C,L	ENN_RA	The range of variation in Euclidean nearest-neighbor distance
128	C,L	ENN_SD	The standard deviation of Euclidean nearest-neighbor distance
129	C,L	ENN_CV	The coefficient of variance of Euclidean nearest-neighbor distance
130	C,L	ECON_MN	The mean of edge contrast index
131	C,L	ECON_AM	The area-weighted mean of edge contrast index
132	C,L	ECON_MD	The median of edge contrast index
133	C,L	ECON_RA	The range of variation in edge contrast index
134	C,L	ECON_SD	The standard deviation of edge contrast index
135	C,L	ECON_CV	The coefficient of variance of edge contrast index

Appendix B

The description of special landscape indices.

Name (Abbreviation)	Formula	Parameters
Patch shape (SHAPE)	$\mathrm{SHAPE(R)} = \frac{0.25 \times p_{ij}}{\sqrt{a_{ij}}}$ $\mathrm{SHAPE(V)} = \frac{p_{ij}}{2\sqrt{\pi a_{ij}}}$	$p_{ij}(m)$: perimeter (m) of patch ij. $a_{ij}(m^2)$: area (m^2) of patch ij.
Unit	**Range**	**Description**
/	≥ 1	Shape index of the patch: When the patch is circular, SHAPE = 1; as the patch shape becomes more complex, SHAPE increases gradually.
Name (Abbreviation)	**Formula**	**Parameters**
Fractal dimension (FRAC)	$\mathrm{FRAC(R)} = \frac{2 \times \ln(0.25 \times p_{ij})}{\ln(a_{ij})}$ $\mathrm{FRAC(V)} = \frac{2\ln(p_{ij})}{\ln(4\pi a_{ij})}$	$p_{ij}(m)$: perimeter (m) of patch ij. $a_{ij}(m^2)$: area (m^2) of patch ij.
Unit	**Range**	**Description**
/	[1,2]	Fractal dimension of the patch: When the patch is circular, FRAC = 1; as the patch shape becomes more complex, FRAC approaches 2.

Name (Abbreviation)	Formula	Parameters
Landscape-shape index in class level (LSI) (class)	$LSI(R) = \frac{0.25 \times \sum_{k=1}^{m} e_{ik}^{*}}{\sqrt{A}}$ $LSI(V) = \frac{\sum_{k=1}^{m} e_{ik}^{*}}{2\sqrt{\pi A}}$	$e_{ik}^{*}(m)$: total length (m) of edge in landscape between patch types (classes) i and k. $A(m^2)$: total landscape area (m^2).
Unit	**Range**	**Description**
/	≥ 1	When the patch type i consists of a single square, LSI = 1; LSI increases as the shapes of patches within the patch type become more irregular.
Name (Abbreviation)	**Formula**	**Parameters**
Landscape-shape index in landscape level (LSI) (landscape)	$LSI(R) = \frac{0.25 \times E^{*}}{\sqrt{A}}$ $LSI(V) = \frac{E^{*}}{2\sqrt{\pi A}}$	$E^{*}(m)$: the total length (m) of edges within the landscape, including boundary edges. $A(m^2)$: total area (m^2) of the landscape.
Unit	**Range**	**Description**
/	≥ 1	The landscape-shape index of patches in the landscape. When the landscape consists of a single square, LSI = 1; LSI increases as the shape of patches in the landscape becomes more irregular or as the total edge length within the landscape increases.

References

1. Forman, R.T.T.; Godron, M. *Landscape Ecology*; Wiley: New York, NY, USA, 1986.
2. Jianguo, W. Landscape ecology-concepts and theories. *Chin. J. Ecol.* **2000**, *19*, 42–52.
3. Ryu, S.R.; Chen, J.; Zheng, D.; Lacroix, J.J. Relating surface fire spread to landscape structure: An application of FARSITE in a managed forest landscape. *Landsc. Urban Plan.* **2007**, *83*, 275–283. [CrossRef]
4. Thiele, J.; Kellner, S.; Buchholz, S.; Schirmel, J. Connectivity or area: What drives plant species richness in habitat corridors? *Landsc. Ecol.* **2018**, *33*, 173–181. [CrossRef]
5. Zhang, R.; Li, S.; Wei, B.; Zhou, X. Characterizing production–living–ecological space evolution and its driving factors: A case study of the chaohu lake basin in China from 2000 to 2020. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 447.
6. Hou, B.; Wei, C.; Liu, X.; Meng, Y.; Li, X. Assessing Forest Landscape Stability through Automatic Identification of Landscape Pattern Evolution in Shanxi Province of China. *Remote Sens.* **2023**, *15*, 545. [CrossRef]
7. Forman, R.T.T.; Godron, M. Patches and structural components for a landscape ecology. *BioScience* **1981**, *31*, 733–740.
8. McGarigal, K.; Cushman, S.A. The Gradient Concept of Landscape Structure. In *Issues and Perspectives in Landscape Ecology*; Cambridge University Press: Cambridge, UK, 2005; pp. 112–119.
9. Bolliger, J.; Wagner, H.H.; Turner, M.G. Identifying and quantifying landscape patterns in space and time. In *A Changing World: Challenges for Landscape Research*; Springer: Dordrecht, The Netherlands, 2007; pp. 177–194.
10. Lausch, A.; Blaschke, T.; Haase, D.; Herzog, F.; Syrbe, R.U.; Tischendorf, L.; Walz, U. Understanding and quantifying landscape structure—A review on relevant process characteristics, data models and landscape metrics. *Ecol. Modell.* **2015**, *295*, 31–41. [CrossRef]
11. Nowosad, J.; Stepinski, T.F. Information theory as a consistent framework for quantification and classification of landscape patterns. *Landsc. Ecol.* **2019**, *34*, 2091–2101. [CrossRef]
12. Huang, C.; Yang, J.; Jiang, P. Assessing impacts of urban form on landscape structure of urban green spaces in China using Landsat images based on Google Earth Engine. *Remote Sens.* **2018**, *10*, 1569. [CrossRef]
13. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
14. Simpson, E.H. Measurement of diversity. *Nature* **1949**, *163*, 688. [CrossRef]
15. Tsallis, C. Entropic nonextensivity: A possible measure of complexity. *Chaos Solitons Fractals* **2002**, *13*, 371–391. [CrossRef]
16. Cushman, S.A. Entropy in landscape ecology: A quantitative textual multivariate review. *Entropy* **2021**, *23*, 1425. [CrossRef] [PubMed]
17. Burrough, P.A. Fractal dimensions of landscapes and other environmental data. *Nature* **1981**, *294*, 240–242. [CrossRef]
18. Alados, C.L.; Pueyo, Y.; Giner, M.L.; Navarro, T.; Escos, J.; Barroso, F.; Cabezudo, B.; Emlen, J.M. Quantitative characterization of the regressive ecological succession by fractal analysis of plant spatial patterns. *Ecol. Modell.* **2003**, *163*, 1–17. [CrossRef]
19. Saura, S.; Estreguil, C.; Mouton, C.; Rodríguez-Freire, M. Network analysis to assess landscape connectivity trends: Application to European forests (1990–2000). *Ecol. Indic.* **2011**, *11*, 407–416. [CrossRef]
20. Foltête, J.C.; Clauzel, C.; Vuidel, G. A software tool dedicated to the modelling of landscape networks. *Environ. Modell. Softw.* **2012**, *38*, 316–327. [CrossRef]
21. de la Barra, F.; Alignier, A.; Reyes-Paecke, S.; Duane, A.; Miranda, M.D. Selecting graph metrics with ecological significance for deepening landscape characterization: Review and applications. *Land* **2022**, *11*, 338. [CrossRef]

22. Merrick, M.J.; Koprowski, J.L. Circuit theory to estimate natal dispersal routes and functional landscape connectivity for an endangered small mammal. *Landsc. Ecol.* **2017**, *32*, 1163–1179. [CrossRef]
23. Liu, X.; Liu, D.; Zhao, H.; He, J.; Liu, Y. Exploring the spatio-temporal impacts of farmland reforestation on ecological connectivity using circuit theory: A case study in the agro-pastoral ecotone of North China. *J. Geog. Sci.* **2020**, *30*, 1419–1435. [CrossRef]
24. Landau, V.A.; Shah, V.B.; Anantharaman, R.; Hall, K.R. Omniscape. jl: Software to compute omnidirectional landscape connectivity. *J. Open Source Softw.* **2021**, *6*, 2829. [CrossRef]
25. Cushman, S.A.; McGarigal, K.; Neel, M.C. Parsimony in landscape metrics: Strength, universality, and consistency. *Ecol. Indic.* **2008**, *8*, 691–703. [CrossRef]
26. Kienast, F.; Frick, J.; van Strien, M.J.; Hunziker, M. The Swiss Landscape Monitoring Program—A comprehensive indicator set to measure landscape change. *Ecol. Modell.* **2015**, *295*, 136–150. [CrossRef]
27. Dong, J.; Dai, W.; Shao, G.; Xu, J. Ecological network construction based on minimum cumulative resistance for the city of Nanjing, China. *ISPRS Int. J. Geo-Inf.* **2015**, *4*, 2045–2060.
28. Li, M.; Luo, G.; Li, Y.; Qin, Y.; Huang, J.; Liao, J. Effects of landscape patterns and their changes on ecosystem health under different topographic gradients: A case study of the Miaoling Mountains in southern China. *Ecol. Indic.* **2023**, *154*, 110796. [CrossRef]
29. Xiao, C.; Wang, Y.; Yan, M.; Chiaka, J.C. Impact of cross-border transportation corridors on changes of land use and landscape pattern: A case study of the China-Laos railway. *Landsc. Urban Plan.* **2024**, *241*, 104924. [CrossRef]
30. Adler, K.; Jedicke, E. Landscape metrics as indicators of avian community structures—A state of the art review. *Ecol. Indic.* **2022**, *145*, 109575. [CrossRef]
31. Frazier, A.E.; Kedron, P. Comparing forest fragmentation in Eastern US forests using patch-mosaic and gradient surface models. *Ecol. Inform.* **2017**, *41*, 108–115. [CrossRef]
32. Chang, S.; Jiang, Q.; Wang, Z.; Xu, S.; Jia, M. Extraction and spatial–temporal evolution of urban fringes: A case study of changchun in Jilin Province, China. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 241.
33. Xu, M.; Niu, L.; Wang, X.; Zhang, Z. Evolution of farmland landscape fragmentation and its driving factors in the Beijing-Tianjin-Hebei region. *J. Clean. Prod.* **2023**, *418*, 138031. [CrossRef]
34. McGarigal, K.; Marks, B.J. *FRAGSTATS: Spatial Pattern Analysis Program for Quantifying Landscape Structure*; Gen. Tech. Rep. PNW-GTR-351; U.S. Department of Agriculture, Forest Service, Pacific Northwest Research Station: Portland, OR, USA, 1995.
35. Yao, Y.; Cheng, T.; Sun, Z.; Li, L.; Chen, D.; Chen, Z.; Wei, J.; Guan, Q. VecLI: A framework for calculating vector landscape indices considering landscape fragmentation. *Environ. Modell. Softw.* **2022**, *149*, 105325. [CrossRef]
36. Bettinger, P.; Bradshaw, G.A.; Weaver, G.W. Effects of geographic information system vector–raster–vector data conversion on landscape indices. *Can. J. For. Res.* **1996**, *26*, 1416–1425. [CrossRef]
37. Wade, T.G.; Wickham, J.D.; Nash, M.S.; Neale, A.C.; Riitters, K.H.; Jones, K.B. A comparison of vector and raster GIS methods for calculating landscape metrics used in environmental assessments. *Photogramm. Eng. Remote Sens.* **2003**, *69*, 1399–1405. [CrossRef]
38. Saura, S.; Torné, J. Conefor Sensinode 2.2: A software package for quantifying the importance of habitat patches for landscape connectivity. *Environ. Modell. Softw.* **2009**, *24*, 135–139.
39. Baker, W.L.; Cai, Y. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. *Landsc. Ecol.* **1992**, *7*, 291–302. [CrossRef]
40. Rempel, R.S.; Kaukinen, D.; Carr, A.P. *Patch Analyst and Patch Grid. Ontario Ministry of Natural Resources*; Centre for Northern Forest Ecosystem Research: Thunder Bay, ON, Canada, 2012.
41. Lang, S.; Tiede, D. vLATE Extension für ArcGIS–Vektorbasiertes Tool zur Quantitativen Landschaftsstrukturanalyse. 2003. Available online: https://uni-salzburg.elsevierpure.com/en/publications/vlate-extension-f%C3%BCr-arcgis-vektorbasiertes-tool-zur-quantitativen (accessed on 3 July 2024).
42. Yu, M.; Huang, Y.; Cheng, X.; Tian, J. An ArcMap plug-in for calculating landscape metrics of vector data. *Ecol. Inform.* **2019**, *50*, 207–219. [CrossRef]
43. Jung, M. LecoS—A python plugin for automated landscape ecology analysis. *Ecol. Inform.* **2016**, *31*, 18–21. [CrossRef]
44. Zaragozí, B.; Belda, A.; Linares, J.; Martínez-Pérez, J.E.; Navarro, J.T.; Esparza, J. A free and open source programming library for landscape metrics calculations. *Environ. Modell. Softw.* **2012**, *31*, 131–140. [CrossRef]
45. Hesselbarth, M.H.; Sciaini, M.; With, K.A.; Wiegand, K.; Nowosad, J. landscapemetrics: An open-source R tool to calculate landscape metrics. *Ecography* **2019**, *42*, 1648–1657. [CrossRef]
46. QGIS. A Free and Open Source Geographic Information System. 2024. Available online: https://www.qgis.org/ (accessed on 3 July 2024).
47. QGIS Development Team. QGIS Geographic Information System. Available online: http://qgis.osgeo.org (accessed on 12 May 2024).
48. PyQGIS Developer Cookbook (QGIS 3.34). QGIS Documentation. Available online: https://docs.qgis.org/3.34/en/docs/pyqgis_developer_cookbook/index.html (accessed on 7 March 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

Enhancing Place Emotion Analysis with Multi-View Emotion Recognition from Geo-Tagged Photos: A Global Tourist Attraction Perspective

Yu Wang [1], Shunping Zhou [2,3], Qingfeng Guan [1,3], Fang Fang [2,3], Ni Yang [1], Kanglin Li [2] and Yuanyuan Liu [2,3,*]

[1] School of Geography and Information Engineering, China University of Geosciences, Wuhan 430078, China; vvy190701@cug.edu.cn (Y.W.); guanqf@cug.edu.cn (Q.G.); yangni@cug.edu.cn (N.Y.)
[2] School of Computer Science, China University of Geosciences, Wuhan 430078, China; zhoushunping@cug.edu.cn (S.Z.); fangfang@cug.edu.cn (F.F.); lkl1202110821@cug.edu.cn (K.L.)
[3] National Engineering Research Center for Geographic Information System, Wuhan 430078, China
* Correspondence: liuyy@cug.edu.cn

Abstract: User-generated geo-tagged photos (UGPs) have emerged as a valuable tool for analyzing large-scale tourist place emotions with unprecedented detail. This process involves extracting and analyzing human emotions associated with specific locations. However, previous studies have been limited to analyzing individual faces in the UGPs. This approach falls short of representing the contextual scene characteristics, such as environmental elements and overall scene context, which may contain implicit emotional knowledge. To address this issue, we propose an innovative computational framework for global tourist place emotion analysis leveraging UGPs. Specifically, we first introduce a Multi-view Graph Fusion Network (M-GFN) to effectively recognize multi-view emotions from UGPs, considering crowd emotions and scene implicit sentiment. After that, we designed an attraction-specific emotion index (AEI) to quantitatively measure place emotions based on the identified multi-view emotions at various tourist attractions with place types. Complementing the AEI, we employ the emotion intensity index (EII) and Pearson correlation coefficient (PCC) to deepen the exploration of the association between attraction types and place emotions. The synergy of AEI, EII, and PCC allows comprehensive attraction-specific place emotion extraction, enhancing the overall quality of tourist place emotion analysis. Extensive experiments demonstrate that our framework enhances existing place emotion analysis methods, and the M-GFN outperforms state-of-the-art emotion recognition methods. Our framework can be adapted for various geo-emotion analysis tasks, like recognizing and regulating workplace emotions, underscoring the intrinsic link between emotions and geographic contexts.

Keywords: place emotion analysis; multi-view emotion recognition; GNN; attraction-specific place emotion indices; user-generated photos

1. Introduction

Place emotions play a pivotal role in shaping human interactions with the environment [1]. Place emotion analysis involves extracting and analyzing human emotions associated with specific locations, allowing us to obtain a wealth of information about scenes, human emotions, behaviors, and their interconnections [2]. Recently, growing interest in this area has been evident, with research focusing on emotional geography and environment perception, such as quantifying park perceptions [3] and analyzing how the forest environment promotes the feelings of people [4]. Moreover, the intersection of intelligent emotion computing and geography in analyzing tourist place emotions has significant implications for tourist behaviors and scenic transformations [5,6].

In recent years, thanks to the advancement of social media, big data technology and emotion recognition [7,8], a sizable quantity of geo-tagged texts and user-generated geo-tagged photos (UGPs) on social networking sites (SNSs) [9] have provided extensive data support for place emotion analysis [10]. UGPs, as opposed to text data, provide a more objective and comprehensive view of emotions, such as facial expressions [11], human behaviors [12], and scene environments [13], without geographical variations, offering a broader perspective for studying emotional aspects worldwide. Consequently, there has been a surge in studies using UGPs to analyze place emotions at different geographic locations. For instance, [14] extracted smiling expressions from photos to describe worldwide geographic emotion patterns. And [15] categorized facial expressions from social media to study city emotion distribution and inter-city emotional similarities. The aforementioned studies offer a foundational theoretical framework for comprehending human emotions in place emotion analysis.

Despite the progress, current methods still pose two-fold challenges. Firstly, the existing methods mainly focus on extracting place emotions through off-the-shelf facial expression recognition (FER) technologies [16,17], overlooking the influence of human emotion relations in crowds, which are crucial for understanding individual behaviors. For instance, a laughing person in a serious parade does not change the overall serious mood. Current FER methods focus on individual faces, especially with posed expressions at tourist sites, often resulting in inaccurate emotion analysis. Secondly, they often neglect geographic and environmental contexts, such as natural landscapes and architectural designs, which are pivotal in determining human emotional responses [18]. Neglecting these factors may create an emotional gap in understanding scene emotions and hinder accurate analysis of place emotions [19].

To address the limitations of existing place emotion analysis methods, we propose an innovative computational framework for analyzing tourist place emotions leveraging UGPs. This framework mitigates the emotional gap by identifying multi-view emotions and understanding attraction-specific types. Through this, our framework offers a more holistic and accurate exploration of affective information, effectively addressing the gap in traditional methods. The primary contributions of this study are the following:

(1) We introduce a unique framework for analyzing tourist place emotions from user-generated photos (UGPs) globally, combining multi-view emotions and attraction types for a thorough emotion analysis at tourist sites;

(2) Our novel M-GFN model goes beyond just facial expressions in UGPs, capturing multi-view emotions, thus suppressing the emotional gap from individual expressions. We also create a new attraction-specific emotion index (AEI) for better emotion analysis in tourism;

(3) To validate our framework's effectiveness and explore the differences and commonalities in place emotions across attraction types, we collected a global dataset of tourist place emotions, called TPE.

2. Related Work

2.1. Place Emotion Extraction

Existing place emotion analysis approaches are categorized into survey-based, natural language processing (NLP)-based, and FER-based methods. Compared with survey- and NLP-based methods, FER-based methods are more practical and objective for automatic place emotion analysis from large-scale UGPs. This section focuses on FER-based methods for analyzing place emotion from UGPs.

FER-based methods extract human emotions from facial expressions, offering advantages of worldwide universality and consistency across cultures for objective emotion analysis, including emotion extraction and analysis [15]. With abundant geo-tagged SNS photos and the achievements of deep learning [20–22], FER allows efficient, large-scale place emotion analysis. For instance, Li et al. [23] extracted human emotions from facial expressions in Flickr photos and mapped the global geographic distribution of human emo-

tion. Svoray et al. [24] utilized Microsoft Emotion API, another FER-based tool, to detect smiles from Flickr photos, exploring human–environment emotion interactions. Despite the potential advantages of FER-based methods for place emotion analysis from UGPs, it is still challenging to directly apply the FER approaches to place emotion extraction. Primarily, existing methods with FER technologies tend to focus on individual facial expressions, often overlooking the posed expressions commonly found in settings like social media [25]. Such expressions might not authentically represent the genuine emotions of individuals. Beyond individual expressions, the mood of a place is largely shaped by crowd behaviors, as well as key scene contextual environmental information like local scene objects, natural landscapes, and weather conditions [26]. However, these crucial aspects are frequently ignored with FER-based methods.

Given these challenges, a holistic framework that integrates individual facial expressions with crowd dynamics and the overarching scene context is imperative for an accurate and comprehensive place emotion analysis.

2.2. Place Emotions in Tourism

Place emotions are tied to place types, influenced by their historical, cultural, or environmental aspects [27]. Different settings, like natural parks and historic monuments, evoke unique emotions [28]. This emotion–place connection is vital for place design to enhance visitor experiences [29].

Building on this idea, emotion is increasingly important in tourism research, as it is essential for tourist experiences [30]. The rise of social media platforms, such as Instagram and Flickr, allows travelers to express and share their emotions during travels. In light of this, Cheung et al. [31] studied the effect of social media-based destination brands on tourist emotions. Mehra et al. [32] used sentiment and emotion analysis to predict tourist behaviors from user-generated comments. Jiang et al. [33] constructed an emotional map of visitors based on emotional experiences and time–space dynamics within attractions.

Despite the importance of emotions in tourism research, there is still a research gap in analyzing the macro emotions of attractions, especially the correlation between different attraction types and place emotions. Different attraction types—natural landscapes or cultural landmarks—likely evoke different emotional responses. Despite this understanding, studies on how specific attraction types affect visitor emotions are still limited. Such an investigation is crucial, as it bridges a significant knowledge gap and thoroughly examines why different attraction types elicit distinct emotions. Filling this gap will provide valuable insights to help optimize tourist experiences and services for specific attractions.

3. Methodology

3.1. Study Area and Datasets

In this section, we first introduce the selected study areas for tourist place emotion analysis and then describe the emotion database used in this study in detail.

3.1.1. Study Areas

In order to mine the emotions of tourist places, 157 unique tourist attractions worldwide were selected for this study. These tourist sites span 38 nations across six continents, encompassing various cultures, place types, regional differences, and natural styles. Figure 1 depicts the distribution of our selected tourist attractions across the world's continents. The specific distribution is as follows: 56 sites in Europe, 49 in Asia, 23 in North America, 12 in Oceania, 11 in South America, and 6 in Africa.

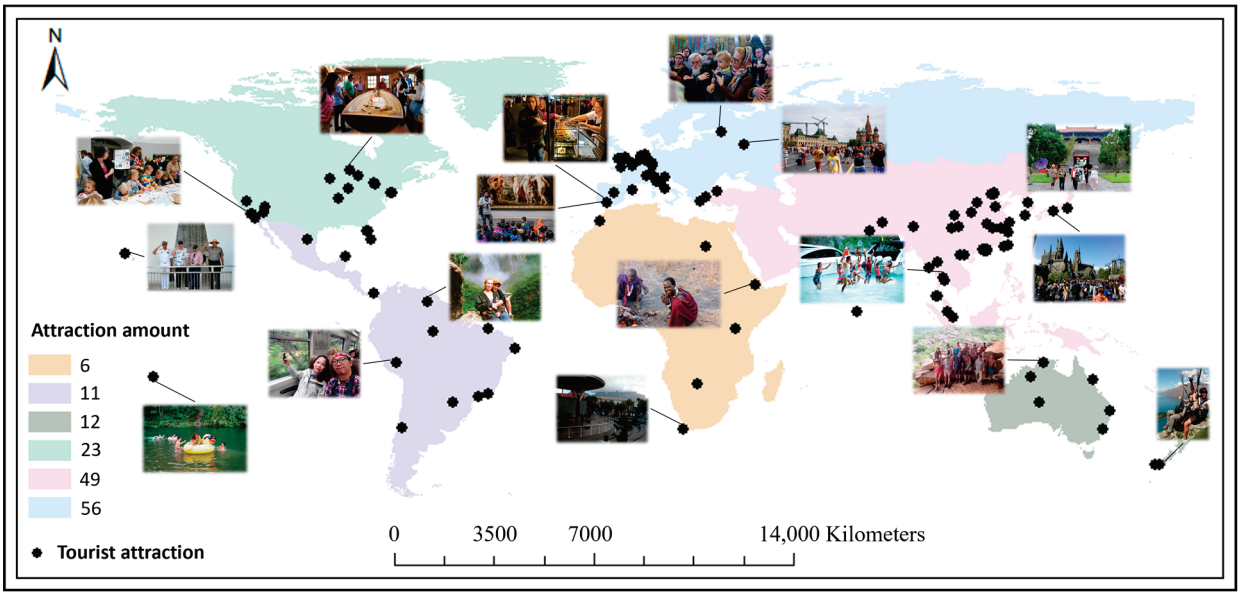


Figure 1. Distributions of selected tourist attractions and UGPs, with colored rectangles indicating attraction numbers across continents.

3.1.2. Datasets

We collected and established a large-scale, geo-tagged tourist place emotion dataset called TPE. TPE offers three types of annotations: scene sentiment category, geo-tagged information, and attraction type. The pipeline of data collection and processing for TPE is shown in Figure 2.

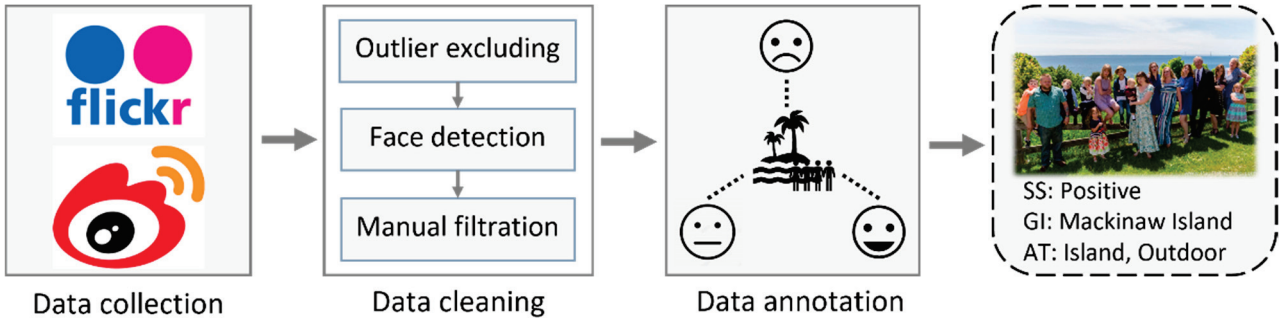


Figure 2. Overview of the collection and the annotation of TPE. Note: SS, GI, and AT are the abbreviations of the scene sentiment, geo-tagged information, and attraction type, respectively.

Data collection. We developed a crawler to collect over 100,000 user-generated photos (UGPs) with the .jpg format from Flickr and Weibo, using tourist attractions' names and coordinates obtained via the Google Geocoding API (https://developers.google.cn/maps/documentation/geocoding/overview),accessed on 1 September 2020. These UGPs come from 38 countries from 6 continents and cover 157 tourist attractions from January 2001 to November 2020. We ensured diversity by collecting at least 1000 images within a 1 km radius of each attraction, prioritizing those rich in human presence and sentiment.

Data cleaning. With the UGPs and the corresponding geo-tagged coordinates, we cleaned the data in both automatic and manual mode. Firstly, we deleted photographs farther than one kilometer from the center of each attraction site. Then, a face detector was employed to detect and save photos with human faces. Thirdly, we manually removed the unqualified data with issues like occlusion (images where key features such as faces were obstructed, either partially or fully), low-resolution (images that did not meet our resolution threshold (e.g., at least 400p) were removed to ensure that all data used could

support accurate facial emotion analysis), mis-detected faces (images where the algorithm incorrectly identified non-face objects as faces were carefully removed to maintain data quality), and so on, and finally obtained 85,022 UGPs with 233,898 faces from various global attractions.

Data annotation. Annotating such a sizeable database is challenging and time-consuming. Unlike other databases that only provide basic expression categories, our database offers three kinds of annotations for UGPs: the scene sentiment category, geo-tagged information, and attraction type. The geo-tagged information consists of the attraction names and the geo-tagged coordinates collected from Google Geocoding API. Figure 3 shows a few cases with compound annotations in our dataset. These three annotations are designed to comprehensively capture the emotional and contextual aspects of the tourist experiences depicted in the images.

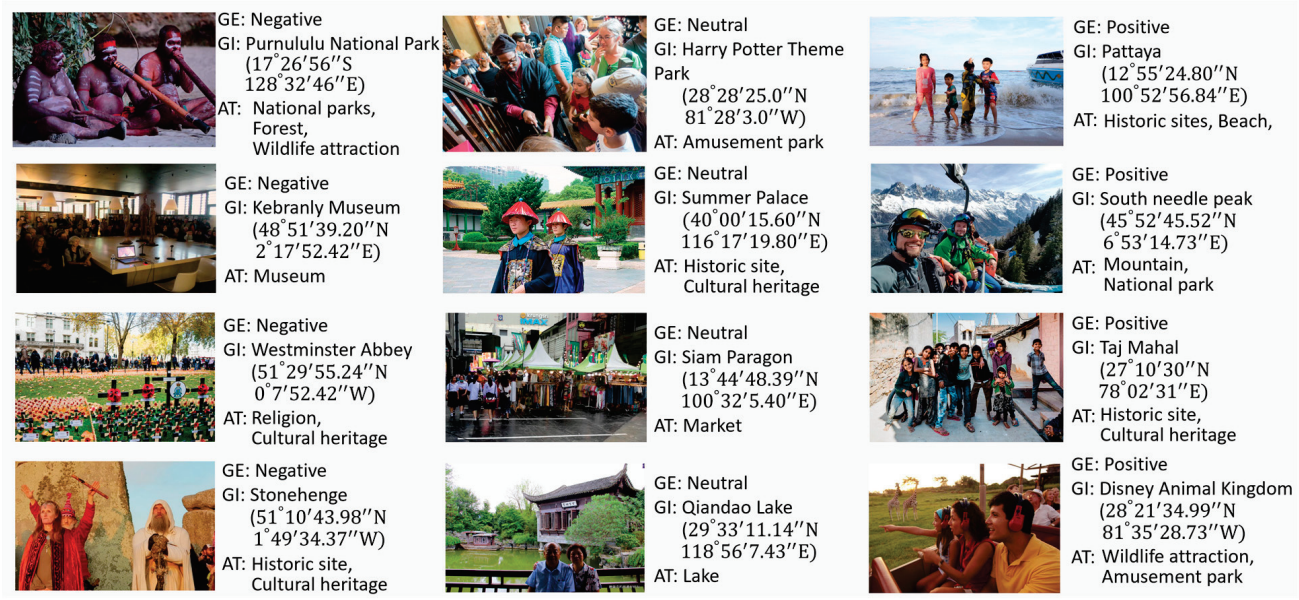


Figure 3. Samples of the TPE at different tourist attractions.

For *scene sentiment category annotation*, we sampled over 15,000 images from the collected emotion images as an annotated TEP subset. To ensure the professionalism of the annotation, we employed five emotional knowledge-trained annotators and developed a software called Expression Label Tool to facilitate efficient annotation. Each UGP in the annotated TEP subset has five working independent annotators. Referring to [34], the UGPs were categorized as positive, neutral, or negative, which reflects the primary emotional responses to the location. For challenging neutral annotation, we rely on simple indicators such as calm facial expressions, scenes without much movement or bright colors, and scenes where people are dressed normally and not interacting much. For positive and negative annotation, we utilize clear emotional expressions and contextual cues: positive emotions are marked by smiles, vibrant colors, and lively interactions, while negative emotions are identified by frowns, dull colors, and signs of stress or conflict. A UGP was kept if most annotators agreed on its category; otherwise, it was discarded for annotation reliability. With a Kappa coefficient [35] of 0.78 indicating high consistency, we finalized a subset of 10,034 UGPs.

For *geotagged information annotation*, each photo in a tourist attraction is annotated with precise geographical coordinates and the name of the attraction, facilitating spatial analysis and correlation with specific locations. This subset was then split into training (8068 images) and testing (1966 images) sets, following existing scene sentiment dataset standards.

For *attraction type annotation*, 157 tourist attractions were categorized into 4 main categories with 15 fine-grained types according to types of attractions listed on Google Travel

(https://www.google.com/travel/), Tourism Teacher (https://tourismteacher.com/), and the *Classification of Tourist Attractions* Group Standards, issued by the China Association of tourist sites on 15 November 2019. Table 1 provides descriptions of the coarse- and fine-grained attraction categories. Inspired by [36], a tourist attraction can be classified into one or multiple types. Like Machu Picchu, it may fit multiple categories (e.g., historical site and cultural heritage). Figure 4 presents the number of UGPs and faces under each tourist attraction type. The type of tourist attraction provides a contextual environment that helps with the analysis of the emotional response to different settings.

Table 1. The classification and description of the attraction types.

Coarse Categories	Fine-Grained Types	Description
Natural landscape	Mountains	Landscapes dominated by naturally occurring mountains, such as Mountain Tai
	Forests	Like Qiandaohu National Forest Park, Halong Bay sea forest, and so on
	Lakes	Such as Lake Superior, Riyuetan Pool, and so forth
	Rivers	Like the Mississippi River, Amazon River and so on
	National parks	Areas are protected because of their diverse fauna and/or lovely surroundings, like Kakadu National Park
	Islands	Lands that are separated from the mainland by water, like Mackinac Island
	Beaches	Areas with pleasant weather and soft sand that attract tourists, like the shores
Cultural landscape	Museums	Places constructed around history, science, culture, or another topic, such as the Metropolitan Museum of Art
	Historic sites	Locations where people go primarily to learn about their histories, like Macchu Picchu
	Religions	Places where people visit mostly for religious-related activities, like cathedrals
	Cultural heritages	Places with the aim of preserving the cultural premises of outstanding universal value to humanity worldwide, like the Great Wall
Purpose-built	Amusement parks	Places built with the sole purpose of providing entertainment for visitors, such as Disneyland
	Wildlife attractions	Areas that enable tourists to see wildlife, such as the zoos and aquariums
Special events	Markets	Destinations where travelers can shop for goods, like the Central Market
	Performance arts	Locations that combine artistic performance and entertainment through forms of cultural expression, like the Paris Opera

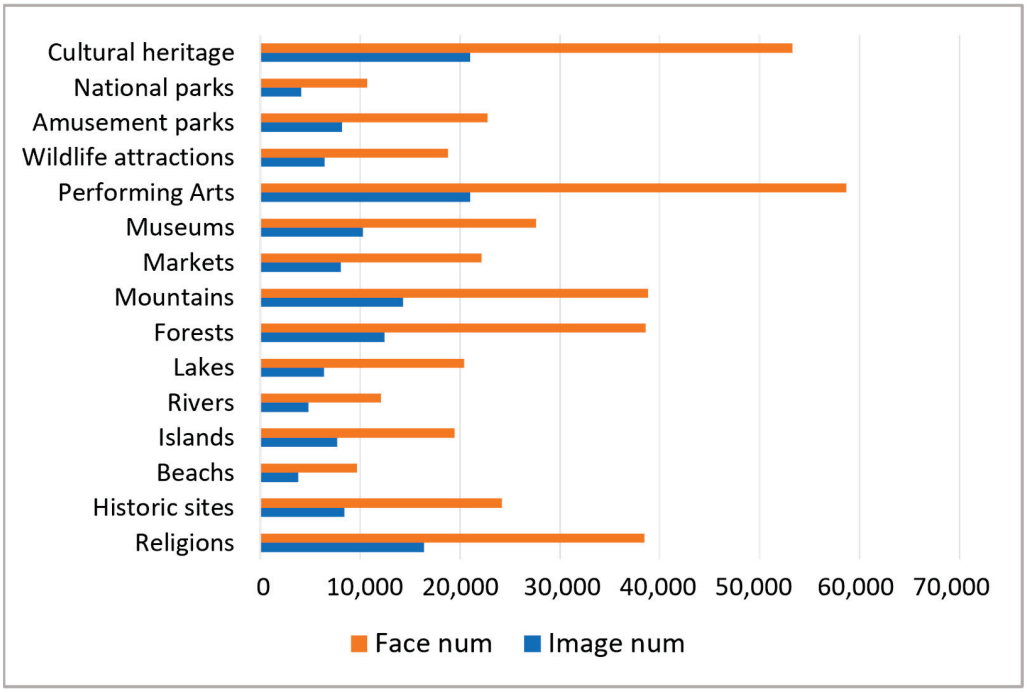


Figure 4. The number of photos (blue) and faces (orange) per tourist attraction type.

In all, we obtain a dataset with broad geographical coverage and cultural diversity, and Table 2 shows a summary of tourist attractions across different global regions, categorized by the number of attractions, the number of countries each region encompasses, and the variety of attraction types available in each region.

Table 2. Global tourist attraction distribution of our dataset.

Regions	Attraction Amounts	Country Amounts	Attraction Type Amounts
Europe	56	10	15
Asia	49	12	15
North America	23	4	15
Oceania	12	2	11
South America	11	3	14
Africa	6	6	12

3.2. Overview

In this section, we present an overview of the proposed tourist place emotion analysis framework for accurate tourist place emotion extraction and analysis around the world, illustrated in Figure 5. The framework contains two main stages: multi-view emotion recognition and attraction-specific place emotion extraction. Particularly, in the first stage, to alleviate the gap between facial expressions and place emotions, we employ the Multi-view Graph Fusion Network (M-GFN) to effectively extract and identify multi-view emotions, including crowd emotions and scene implicit sentiments. In the second stage, with the recognized crowd emotions and scene sentiments from UGPs of different attractions, we design the novel attraction emotion index (AEI), alongside existing indices, the emotion intensity index [2] (EII) and Pearson correlation coefficient [37] (PCC), to facilitate a comprehensive analysis of place emotions at different tourist attractions. With both stages, we can obtain objective and accurate tourist place emotions from a large amount of UGPs.

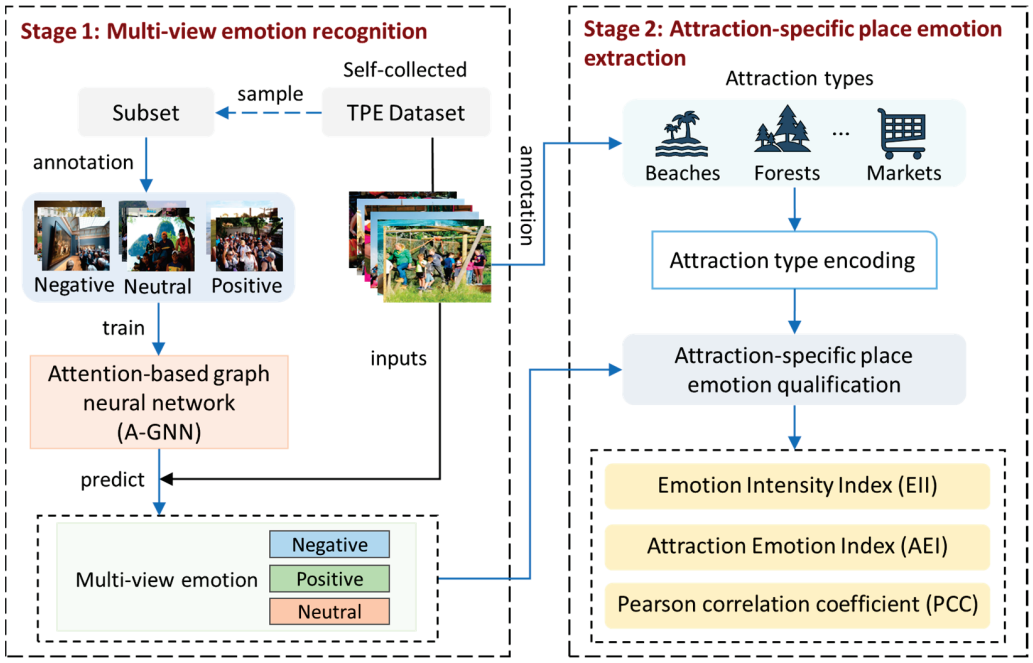


Figure 5. Overall research workflow. It consists of two stages: (1) Training M-GFN on annotated TPE subset for multi-view emotion identification in UGPs from various attractions. (2) Integrating multi-view emotions with attraction types and evaluating place emotions using AEI, EII, and PCC.

3.3. M-GFN for Multi-View Emotion Recognition

Due to the tendency of users to post positive emotions on line [2], UGPs at tourist attractions may contain fake smiles and posed expressions, creating a gap between visible facial expressions and real place emotions. To address this, instead of only identifying individual facial expressions, we recognize multi-view emotions including crowd expressions and scene sentiments from UGPs through a novel M-GFN. This method fully considers the relations among facial expressions, crowd behaviors, and scene environments under different tourist attraction types. The M-GFN process (Figure 6) involves multi-view emotion representation and multi-view emotion recognition.

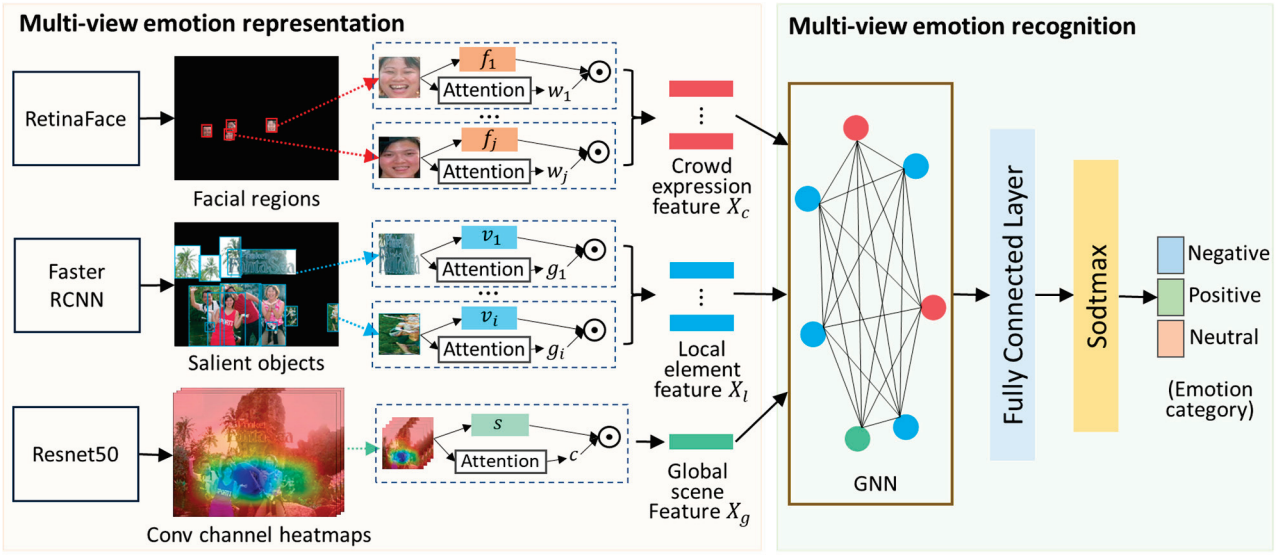


Figure 6. Pipeline of multi-view emotion recognition from attraction UGPs. Given a UGP, the M-GFN first employs hierarchical attention modules to extract multi-view emotion representation and then employs the GNN to accurately identify crowd emotions and scene sentiments.

3.3.1. Multi-View Emotion Representation

In order to obtain multi-view emotion information from UGPs at different attractions, we first extract the features of facial expressions, attraction-specific environmental elements, and the scene.

In particular, with the help of normal DCNNs, such as a pre-trained RetinaFace [38], Faster RCNN [39], and ResNet50 [40], we apply three-level parallel attention augmentations for extracting multi-view environmental emotion characteristics. The first level extracts facial-level crowd expression attention, the second focuses on object-level local attention, and the third learns scene-level global attention.

For the *first-level* attention, we first use the off-the-shelf face detector RetinaFace and pre-trained Resnet50 to detect and extract the facial feature f_j of each individual face region and then apply an LSTM to learn the attention weight of each face, which is able to model the expression relations between different faces in UGPs. Formally, the crowd expression features *Xc* of multi-faces can be calculated as

$$X_c = LSTM(f_1, f_2, \cdots, f_N) = \left[w_1 f_1, w_2 f_2, \cdots, w_N f_{N_1}\right] \in \mathbb{R}^{L_1 \times N} \quad (1)$$

where $LSTM(\cdot)$ represents the overall operation of LSTM, N represents the number of detected faces in a UGP, w_j is the learned attention weight of the j-th facial feature, and L_1 is the dimension of each facial feature vector, indicating the relevance of each face in contributing to the overall crowd emotion. The attention weights of each facial element are dynamically learned through the LSTM's pretraining process.

For *second-level* attention, Faster RCNN is first used to extract different types of local environmental elements in UGPs, like transportation, urban infrastructure, outdoor activities, animals, food and dining, personal items, indoor furniture, and people, as illustrated in Table 3. Then, we introduced a gated attention network, namely SE-Resnet50 [41] pre-trained on the ImageNet-1K database [42], to further identify and extract the features of tourism environmental elements in different attractions. The second-level feature stream, i.e., local attraction—specific environmental element features X_l, can be expressed as

$$X_l = S(e_1, e_2, \cdots, e_M) = [g_1 v_1, g_2 v_2, \cdots, g_M v_{\mathrm{M}}] \in \mathbb{R}^{L_1 \times N} \quad (2)$$

where v_i is the i-th element feature of e_i, g_i is the attention weight learned by the gated attention network, M is the number of salient objects in the UGP, and L_2 is the dimension of each element feature vector. The attention weights are dynamically learned through the gated attention mechanism of SE blocks within the SE-ResNet50.

Table 3. Local environment list.

Environment-Related Categories	Subdivided Local Environment Elements
Transportation	Bicycle, car, boat, bus, etc.
Urban infrastructure	Traffic light, fire hydrant, parking meter, etc.
Outdoor activities	Kite, skateboard, frisbee, sports ball, etc.
Animals	Bird, elephant, giraffe, horse, etc.
Food and Dining	Dining table, sandwich, hot dog, cup, etc.
Personal items	Backpack, handbag, umbrella, suitcase, etc.
Indoor furniture	Chair, sofa, potted plant, vase, etc.
People	Person

For *third-level* attention, to further utilize the global scene information, we first extract global scene features using the pre-trained ResNet50 and apply channel attention to focus on scene features most relevant to the overall sentiment of a UGP. This channel attention mechanism dynamically determines the attention weights, denoted as w_c, by evaluating the relevance of each channel in contributing to the scene sentiment. Then, we use element-wise multiplication to obtain the attraction-specific scene feature $X_g = s \cdot w_c \in \mathbb{R}^{L_3}$, and L_3 is the dimension of the attraction-specific scene feature.

3.3.2. Multi-View Emotion Recognition

With the obtained X_c, X_l, and X_g, we first use a rectified linear unit (ReLU) operation to embed them into a unified vector space to obtain the environmentally-aware scene sentiment representation,

$$X_s = ReLU\left[X_c, X_l, X_g\right] \quad (3)$$

Then, we build the attraction-specific scene graph $G(X_s,\ M_s)$ based on the global scene feature. $M_s = \{m_i\}$, is the aggregate messages set, node vector x_i collects messages from neighboring nodes $\left\{x_j\right\}$ to form $m_i = W_i x_j + b_i$, and W_i and b_i are the weight parameter and the bias vector of the network, respectively. Using the constructed graph $G(X_s, M_s)$ as input, 4-layer Gate Recurrent Units (GRUs) are also used to update node vectors in graph $G(X_s,\ M_s)$ until convergence. After training, a simple full-connected layer followed by the Softmax is used to predict the final scene sentiment.

3.4. Attraction-Specific Place Emotion Extraction

Leveraging identified multi-view emotions, we further develop AEI with two indices, EII and PCC, for a comprehensive analysis of place emotions across global tourist attractions. This emotion extraction involves two phases: attraction type encoding and attraction-specific place emotion quantification (Figure 7).

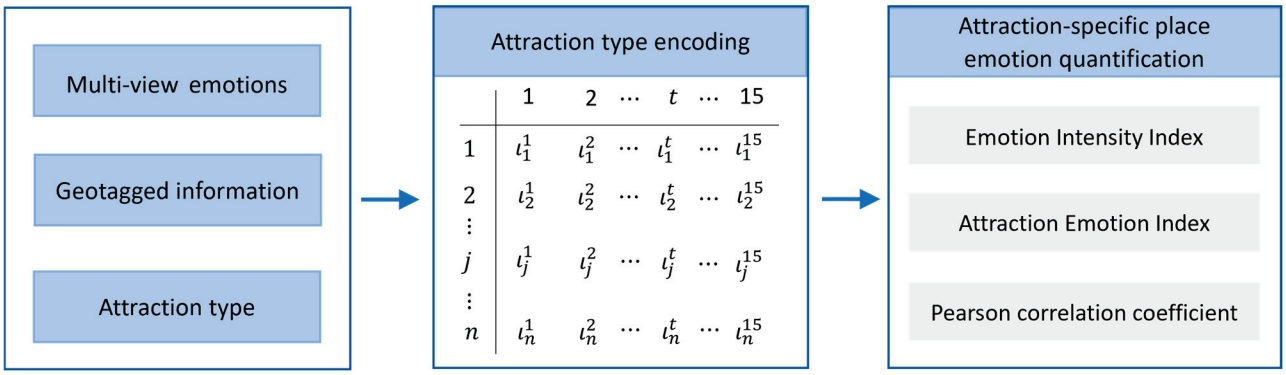


Figure 7. Block diagram of attraction-specific place emotion extraction.

3.4.1. Attraction Type Encoding

Several studies suggested that tourist attraction types may have a significant impact on human emotions. For example, attractions with water features often evoke joy, while concentration camp memorials typically induce neutral or negative emotions. Therefore, tourist attraction types need to be considered in the context of place emotion extraction from tourist places. Following the existing works [43,44], we define 15 types of tourist attractions as detailed in Table 1 in Section 3.1.2.

For each tourist attraction j, we construct a multi-label vector $I_j = \left\{\iota_j^1, \iota_j^2, \cdots, \iota_j^t, \cdots, \iota_j^{15}\right\}$ to encode their types. Where each element ι_j^t represents whether or not the attraction type t is included. Given each attraction $j \in \{1, 2, ..., n\}$, where n = 157, we label the attraction type label ι_j^t in a one-hot encoding manner as follows:

$$\iota_j^{\mathrm{t}} = \begin{cases} 1, & if\ j\ belongs\ to\ t \\ 0, & otherwise \end{cases} \tag{4}$$

where $\iota_j^t = 1$ represents that the attraction j is classified into the attraction type t.

3.4.2. Attraction-Specific Place Emotion Quantification

Using the recognized multi-view emotions and the attraction type encoded vector, we mine the attraction-specific place emotions via these three proposed indices, namely EII, AEI, and PCC.

Emotion Intensity Index (EII). EII quantifies the emotional intensity of each scene sentiment category across attraction types. With a certain attraction type t, EII_t^c is calculated as the ratio of the number of photos in category c to the total number of photos in that type, symbolized as:

$$EII_t^c = \frac{n_t^c}{n_t^{all}} \tag{5}$$

where n_t^c is the number of photos with emotion category c, and n_t^{all} is the total photo count for attraction type t. Note that the closer the EII is to 1, the stronger the intensity of the emotion category c.

Attraction Emotion Index (AEI). AEI is a novel metric for quantifying emotions across different tourist attraction types. It incorporates multi-view emotions, crowd expressions, and scene sentiments. In one tourist attraction, we suppose that the total number of photos is N, the crowd facial expression value of each photo i is fe_i, and C_p and C_n are the counts of photos with positive and negative sentiments, respectively. AEI under each tourist attraction j is represented as AEI_j, which can be calculated as follows:

$$AEI_j = \frac{C_p - C_n}{C_p + C_n} \tag{6}$$

A higher value represents that more people are happy or surprised at that place, which represents positive place emotion. On the contrary, a lower value indicates that more people may be sad, solemn, etc. This might point to a gloomy atmosphere there.

Pearson Correlation Coefficient (PCC). PCC assesses the correlation between place emotions and attraction types, potentially reflecting geographical influences. With AEI_j of the attraction j and the type encoding ι_j^t of the attraction type t, we represent the PCC as $P_{t,AEI}$, which is expressed as follows:

$$P_{t,AEI} = \frac{\sum_{j=1}^{n}\left(AEI_j - \overline{AEI}\right)\left(\iota_j^t - \bar{t}\right)}{\sqrt{\sum_{j=1}^{n}\left(AEI_j - \overline{AEI}\right)^2}\sqrt{\sum_{j=1}^{n}\left(\iota_j^t - \overline{\mathrm{I}^t}\right)^2}}, \tag{7}$$

where n denotes the number of attractions, $\overline{AEI}$ represents the mean of the AEI_j across attractions, and $\overline{\iota^t}$ represents the mean of the attraction type elements. A positive PCC value indicates that the type of tourist attraction has a positive effect on the emotion index AEI, and vice versa.

4. Results and Analysis

In this section, we thoroughly evaluated and discussed our tourist place emotion analysis framework. We first evaluated the results of multi-view emotion recognition. Then, we confirmed the viability of the tourist place emotion analysis framework and explored the connection between place emotions and attraction types, as well as the emotional differences among different tourist attraction types.

4.1. Performance of M-GFN

4.1.1. Heatmap of Multi-View Emotion Representation

To demonstrate the criticality of our three attentions more intuitively, we visualized the heatmaps for crowd expression, local attraction-specific environmental element, and global scene features in Figure 8, respectively. In the hot spring scene, the heatmap on crowd facial features primarily illuminates the faces of the individuals, showcasing the model's precision in identifying expressions in a positive setting. The local heatmap emphasizes handheld items like chairs and slippers, indicating a focus on smaller, significant objects within the interaction. The global scene heatmap outlines the main relaxation area occupied by the group. The second row features a crafting scene where the heatmaps illustrate faces and local items like pens, cups, and plants, with the global heatmap covering the entire activity area. In the park scene, the facial features heatmap focuses on the three individuals, capturing their expressions. The local features heatmap highlights nearby elements such as the bag and roof, while the global heatmap encompasses the surrounding park area. In all, this three-level approach illustrates the model's adaptability in shifting focus according to scene content, adeptly analyzing various aspects of the scene to enrich understanding of complex environments.

4.1.2. Multi-View Emotion Recognition Results

To demonstrate the superiority of the proposed M-GFN for multi-view emotion recognition, we compared our method in terms of accuracy [7] with several deep learning models of Resnet50 [40], SE-ResNet50 [41], context-aware emotion recognition network (CAER-Net) [45], Long Short-Term Memory (LSTM) [46], Vision Transformer (ViT) [47], Graph neural network (GNN) [34], and a self-fusion network [48] based on contrastive learning (SFN). In practice, we used the annotated TEP subset and a popular scene emotion recognition dataset GroupEmoW [34] for comprehensive evaluation. Specifically, we utilized 8068 images from the annotated TEP subset for training, with the remaining 1966 images for testing. For the GroupEmoW, 12,714 images were used for training and 3178 for testing, as summarized in Table 4.

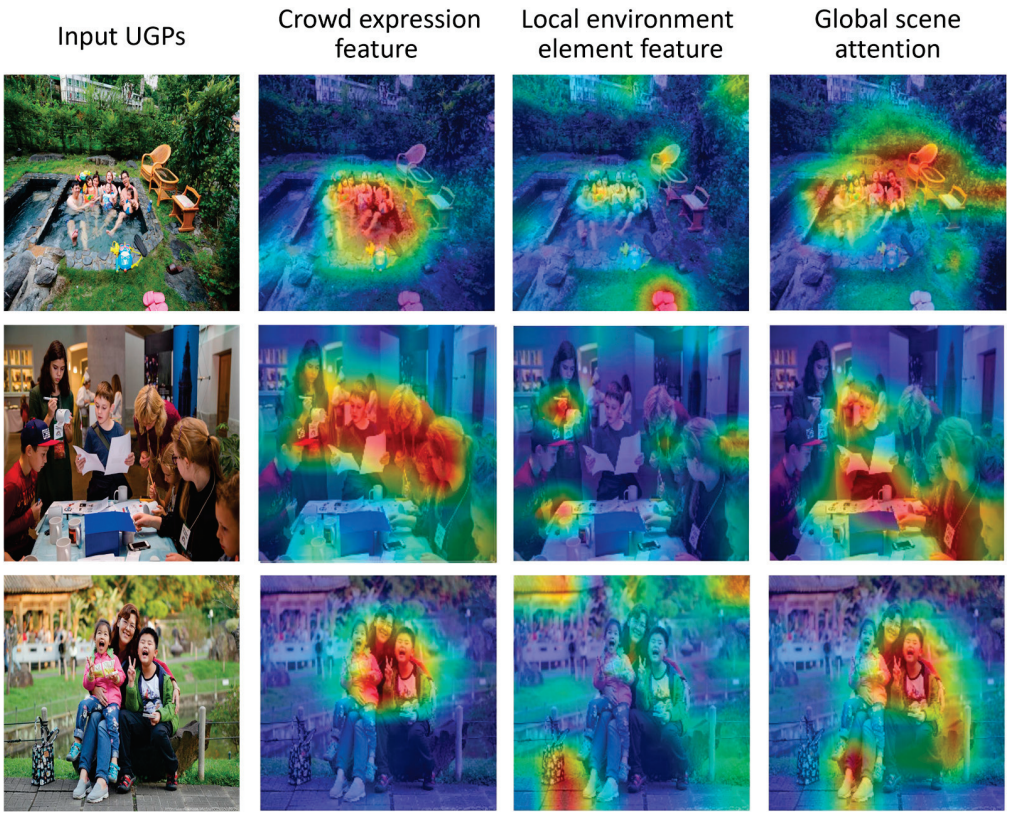


Figure 8. Heatmaps of three-level attentions on sample UGPs.

Table 4. Configuration of sample numbers.

Dataset	Annotated TEP Subset	GroupEmoW
Training	8068	12,714
Testing	1966	3178

All experiments were implemented on Pytorch and TensorFlow libraries and ran on a PC with an Intel Core i7-10700 CPU at 2.90 GHz, 16 GB memory, and a NVIDIA GeForce GTX 2070 SUPER. For training, the learning rate was initialized as $2e^{-4}$ dropped by a factor of 10 per 4 epochs. The findings are summarized in Table 5. Compared to the second-best GNN method, our proposed M-GFN improved the recognition accuracy by 6.18% and 0.7% on the annotated TEP subset and GroupFmoW, respectively. This suggested that the M-GFN is more suitable for multi-view emotion recognition for such tasks.

Table 5. Quantitative evaluation of M-GFN versus other methods on two datasets. The best results are in bold.

Methods	Accuracy (%)	
	Annotated TEP Subset	GroupFmoW
Resnet50 [40]	62.64	71.58
SE-Resnet50 [41]	67.57	69.79
CAER-Net [45]	68.98	80.61
LSTM [46]	69.71	82.76
VIT [47]	68.82	83.47
GNN [34]	75.17	84.62
SFN [48]	68.73	84.15
M-GFN (proposed in this study)	**81.35**	**85.32**

4.2. Tourist Place Emotion Mapping and Analysis

In the section, we evaluated and discussed the effectiveness of the extracted tourist place emotions on the following two aspects: (1) the distribution of the attraction emotion index (AEI) in different continents, (2) analyzing the emotion intensity index (EII) under different attraction types, and (3) exploring the relations between the attraction types and tourist place emotions with the Pearson correlation coefficient (PCC).

4.2.1. Attraction Emotion Index Mapping

The spatial distribution of all 157 tourist attractions with their AEI values is illustrated in Figure 9, with color-coded circles indicating varying AEI levels. Darker blue circles signify more positive place emotions, while lighter blue suggests lower emotional values. The Basilica of Our Lady of Guadalupe has the lowest AEI of 0.292, because religious attractions may inspire solemn emotions. In contrast, Mackinac Island obtains the highest AEI of 0.875, and Orchard Road, Singapore's famous shopping street, obtains an AEI of 0.644, highlighting that a beautiful and relaxing environment promotes a greater AEI. Additionally, an AEI average across continents shows Europe with the lowest at 0.601, likely due to its abundance of historical sites, aligning with findings of recent humanities studies [49]. The average AEI for African attractions is the highest, at 0.798. The high emotional value associated with attractions in Africa could be attributed to its beautiful natural landscapes, diverse wildlife, rich cultural heritage, adventure activities, and high-quality tourism services, offering visitors a uniquely enriching experience. This further demonstrates the relationship between the attraction types and specific cultural contexts, validating our framework.

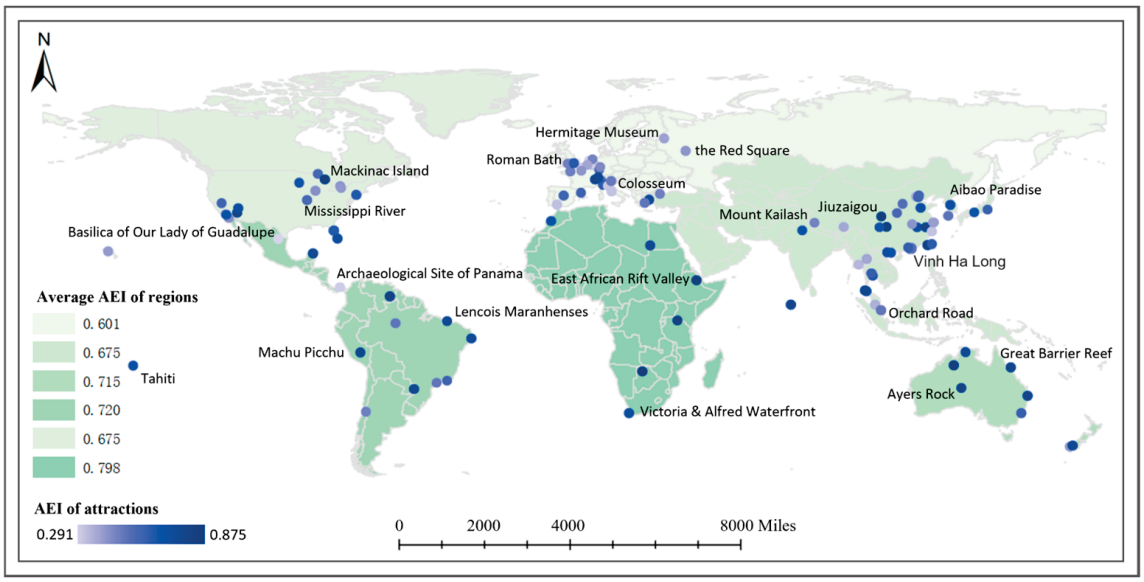


Figure 9. The distributions of AEI for 157 tourist attractions in different continents.

4.2.2. Emotion Intensity Index Cross Attraction Types

Figure 10 provides the EII of the three classes of emotions under each tourist site type. It reveals varying emotional intensities among different tourist attractions. Notably, places like beaches, islands, lakes, forests, mountains, and national parks tend to evoke more positive emotions, often exceeding 50%. The UGPs from those tourist attractions usually contain open environments, beautiful views, or entertainment programs. In contrast, attraction types such as religious sites, historical sites, cultural heritages, and museums have significantly higher EII for negative emotions. This could be because these places often evoke a sense of reflection and mourning in people. Furthermore, neutral emotion constitutes a significant proportion in most cases. Neutral emotion predominates, often reflecting a baseline state when individuals process new information, as seen in tourists

initially absorbing their surroundings [50]. Instead, it can represent a state of contemplation, absorption, or simply a momentary pause in emotional display.

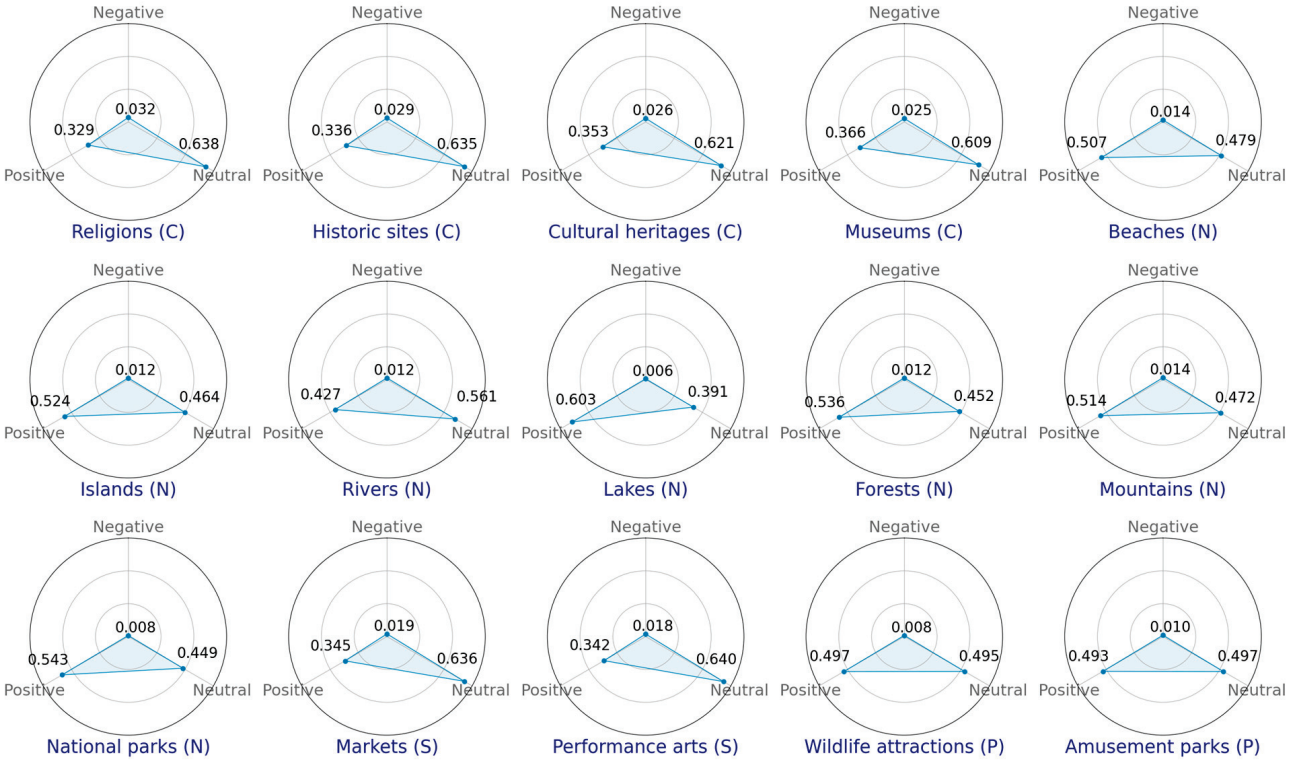


Figure 10. EII of the three classes of emotions for 15 fine-grained tourist attraction types.

4.2.3. Correlation of Attraction Types

To explore the relationship between different attraction types and emotions from the UGP, we calculated the cross-correlation matrix, as illustrated in Figure 11. The four coarse-grained types of tourist attractions have been marked: N (natural landscape), C (cultural landscape), P (purpose-built), and S (special events). Overall, there is a positive correlation among attraction types of religious sites, historic sites, cultural heritages, and museums. These tourist attraction types are all cultural landscapes that will elicit negative emotional reactions from participants. Moreover, there is a strong positive connection between islands and beaches. This could be due to their frequent co-occurrence as travel destinations, with many islands featuring their own beaches. Lakes, forests, mountains, national parks, and wildlife attractions also exhibit positive correlations due to their shared association with natural landscapes and outdoor recreational activities, which are commonly sought after in eco-tourism and adventure travel. On the contrary, there is a negative correlation between cultural landscapes (i.e., religious sites, historic sites, cultural heritages, and museums) and all other attraction types, except for performance arts. Specifically, the result indicates a weak positive association between performance arts and cultural heritages ($r = 0.06$) and museums ($r = 0.01$), suggesting the subtle connections between different types of cultural and artistic sites. For the correlations between the attraction emotions and attraction types, aside from cultural landscapes, all types of attractions have a positive impact on emotions. This finding is consistent with [49], which evaluated human emotions using faces in social media data.

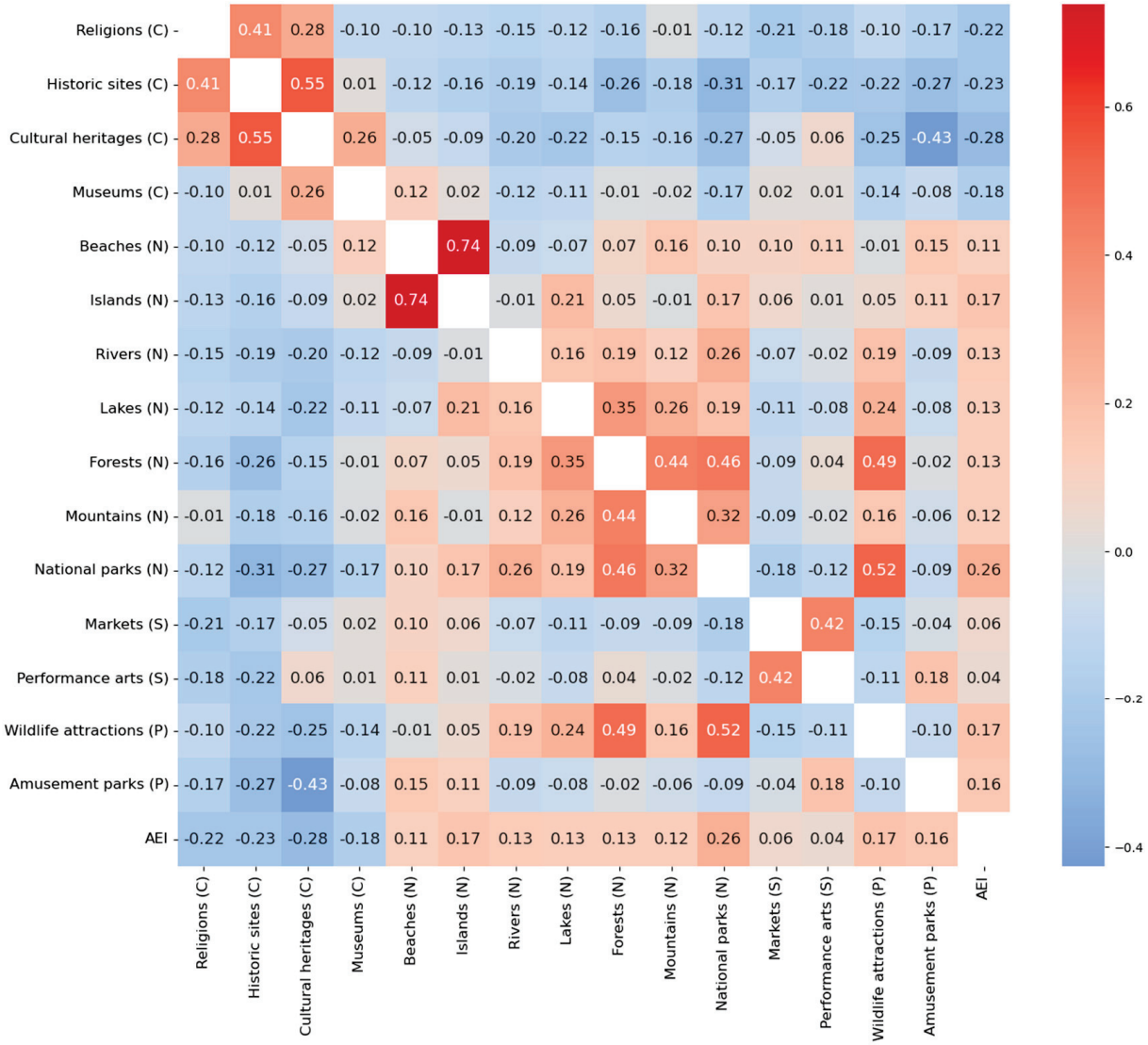


Figure 11. Cross-correlation between the tourist attraction types and AEI.

4.2.4. Predicted Emotions of UGPs under Attraction Types

To further explain how the multi-view emotions of UGPs reflect the place emotions, Figure 12 shows the predicted results of our multi-view emotions and place emotions under different tourist attraction types. The rightmost shows the EII and PCC values of different attraction types, and the left shows the predicted multi-view emotions. In the first row for the attraction type of national parks, the EII for negative emotions is only 0.008, the lowest compared to other attraction types, and the PCC is 0.260, indicating that national parks have a positive effect on the AEI. In the second row, under the wildlife attractions, the multi-view emotions of UGPs show that tourists have positive and neutral emotions. Compared to national parks, the EII values of neutral emotions increase to 0.495 and PCC decreases to 0.175. For the third row of historical sites, the EII values of negative emotions increases, and PCC decreases to -0.226, indicating a negative effect of this attraction type on AEI. The predicted results of different UGPs explicitly corroborate that the emotions of different scenarios constitute place emotions and create place emotions.

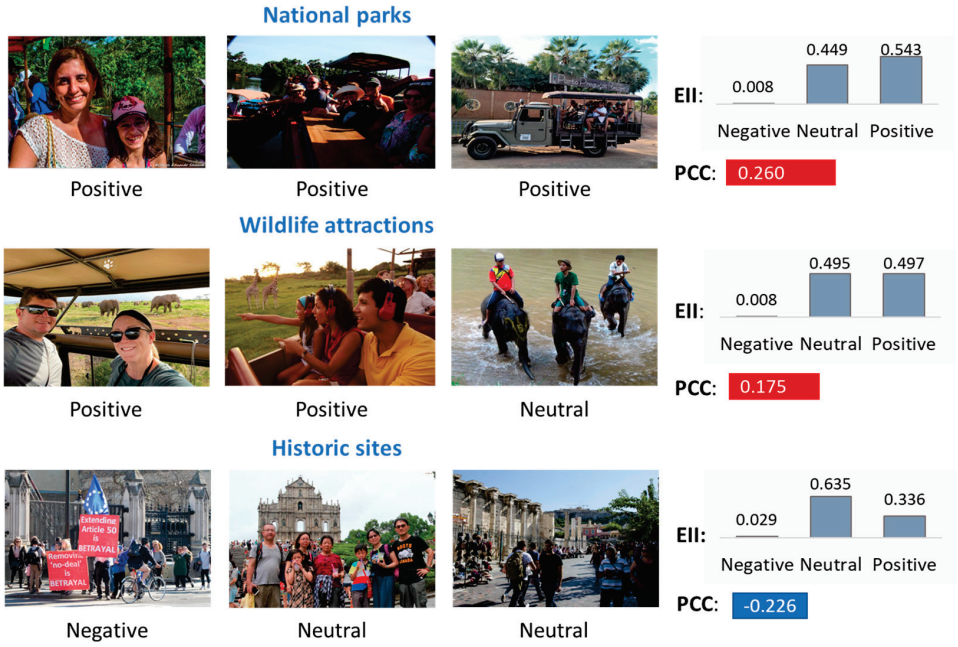


Figure 12. Predicted results of multi-view emotions, EII, and PCC under different attraction types.

4.3. Comparison with Facial Expression-Based Method

4.3.1. M-GFN on Complex Scenarios

Furthermore, to evaluate our method on complex scenes and images with ambiguous expressions, Figure 13 displays the emotion recognition results predicted with the face-based GNN method [34] and our M-GFN. These complex environments contain a variety of crowded scenes, lighting variations, pose variations, occlusion variations, resolution variations, cultural differences, and so on. In contrast to the face-based GNN method [34], our M-GFN demonstrates superior performance. For instance, although the individuals in the first photo of the first row are smiling, the M-GFN accurately identified the prevailing neutral emotion. These visualization results highlight the effectiveness of the M-GFN in suppressing emotion bias among UGPs, ensuring more accurate emotion recognition even in complex scenarios with mixed emotional cues. Unlike traditional models that rely solely on facial expressions, our M-GFN method integrates multiple sources of emotional data, resulting in more robust emotion recognition capabilities.



Figure 13. The emotions predicted with the face-based GNN method and the proposed M-GFN on complex scenes. Obviously, our M-GFN achieved a more robust performance.

4.3.2. Assessment of the Framework

Recognizing the established reputation and adoption of facial expression-based emotion detection in numerous studies, we aimed to validate the reliability of our novel methodology by juxtaposing it against this conventional approach. For this comparative analysis, referring to Kang [49]'s study, which used facial expression tools to create a 'Joy Index', we found our AEI strongly correlates (0.796, p-value: $2.747e^{-21}$) with this index, validating our method's reliability.

Furthermore, unlike traditional methods centered on facial expressions, which at times may misunderstand certain facial cues, our approach offers a more comprehensive insight. Relying only on facial expressions can overlook critical contextual information. Factors like scene objects and surrounding events can profoundly influence emotions, yet these are often disregarded in conventional methods. Our method, incorporating multi-view emotions from the M-GFN, aligns with environmental psychology principles by including such context, offering a more comprehensive and objective emotion understanding. Specifically, results from our multi-view emotion-based framework in Figure 14 reveal that natural and recreational sites like national parks and beaches have a positive impact on human emotions. Purpose-built sites of markets and performance arts also have some positive impact on human emotions. These places often provide experiences that connect individuals with nature, offer entertainment, and facilitate relaxation and physical activity, which can lead to increased happiness and reduced stress. In contrast, cultural landscapes of museums, religious sites, historic sites, and cultural heritages might have a negative impact on emotions. Visitors to these places often reflect on their histories and cultures, evoking solemn and reflective emotions. While results from the individual facial expressions-based framework on the TPE dataset show that amusement parks, markets, and performance arts have a negative connection to human emotions, our findings align more closely with common-sense expectations in our study area.

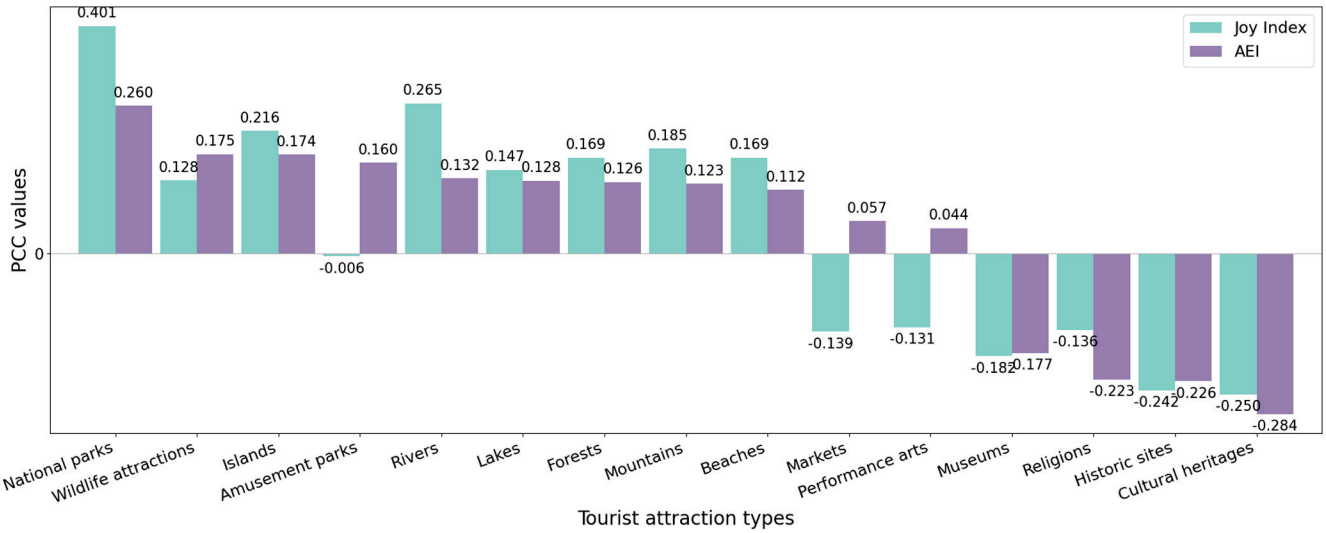


Figure 14. PCC of attraction types and place emotions. AEI via recognized multi-view emotions; Joy Index through single facial expression.

By considering the environment's influence on emotions, our approach captures a wider range of emotional responses (e.g., facial expressions, scene elements, and contexts), making it suited for assessing complex emotional experiences at tourist attractions, indicating the importance of environmental perception of tourist destinations in shaping emotions [51].

4.4. Comparison with Website User Attraction Ratings

We employed the predicted attraction emotion indexes (AEI) that are the emotion experience of people at tourist attractions obtained from the UGPs, for evaluating the results of place emotion prediction. To verify the effectiveness and sensibility of the framework of our proposed place emotion prediction, we compared the predicted AEI and the user rating score in Tripadvisor. The graph in Figure 15 demonstrates remarkable consistency between our normalized emotional analysis results (orange points) and user ratings (blue points), highlighting the effectiveness of our emotion analysis methodology. Additionally, the advantage of our method is that it allows for a more nuanced understanding of visitor emotion responses beyond simple numerical ratings, enabling attractions to tailor experiences and marketing strategies more effectively to meet visitor expectations and enhance overall satisfaction.

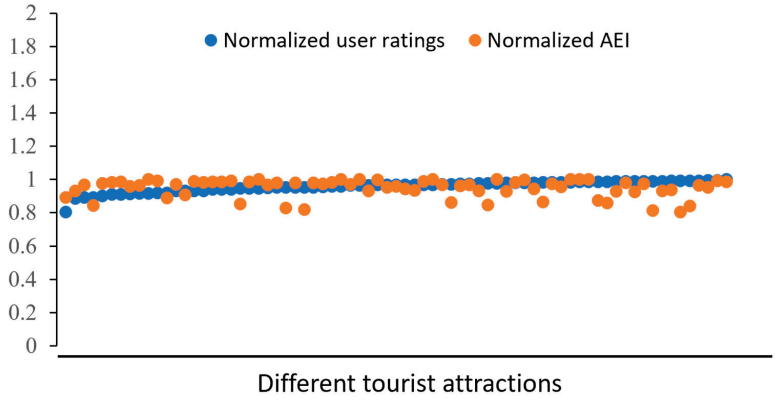


Figure 15. Scatterplot of predicted AEI and rating score of places.

5. Discussion

5.1. Data Sensitivity Test

Since the photos varied across different tourist attractions, a key issue is that more popular attractions may dominate the analysis and skew the findings. We employed a strategy that involves calculating the emotion intensity index (EII) in three data setting scenarios: using data from all attractions, excluding attraction with the most photos, and excluding attraction with the fewest photos. Additionally, we plotted the deviations of the three sets of results in Figure 16 and found that the variance was minimal. Notably, for attraction types like beaches, islands, markets, performance arts, and amusement parks, the variances are still small, all below 0.00015, indicating consistent findings across these three scenarios. This validates and confirms the robustness of our collected data and the place emotion analysis framework in addressing this potential bias.

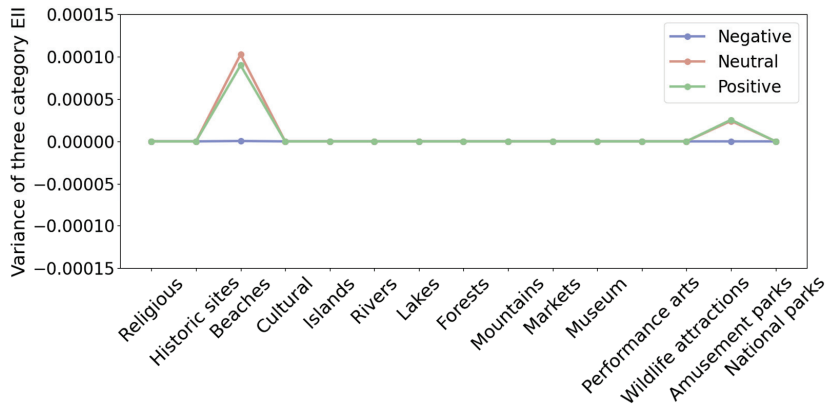


Figure 16. Variance of EII across attraction types with different data settings.

5.2. Impact of Attraction-Type Perspective on Emotions

5.2.1. Differences in Place Emotions across Attraction Types

Due to the introduction of attraction type encoding, we observed that the mined emotional patterns of different tourist places are consistent with existing research, again validating the effectiveness of our framework. For example, Figure 10 further reveals that natural landscapes evoke positive emotions, while cultural sites like religious sites and museums may suppress positive expressions, as Kang et al. indicated [49].

Tourist place emotions are influenced by environmental and human geographical factors. Natural landscapes often induce positive feelings due to their tranquility and inherent beauty, providing a break from urban life [13]. Conversely, cultural sites in regions may prompt more deep emotions, influenced by societal norms and the environments of historical or religious sites [52]. These interactive factors offer a comprehensive perspective on the diverse emotional responses across various attractions.

5.2.2. Commonalities in Place Emotions across Attraction Types

We also observed a notable consistency in the emotional analysis results across both the overall datasets and various attraction types. The negative emotions accounted for the least, less than 5%. This aligns with the previous study [2], which found people may be more inclined to share positive experiences than negative ones on the public platforms. This consistency strengthens our observations.

This finding also emphasizes that, regardless of the type of attraction, they are generally attractive and positively affect tourists. Attractions, as an important part of tourism, are usually designed and managed to provide pleasurable and memorable experiences that generate positive emotions among tourists and reduce the generation of negative emotions [53,54].

To conclude, multiple factors influence the emotional responses of tourists at different attraction types, and a deeper understanding of these reactions could offer more comprehensive insights into tourist behavior.

5.3. Place Emotion Analysis under Various Regions

Understanding the impact of regional culture differences on place emotions is crucial for a comprehensive analysis of tourist experiences. Figure 17 illustrates the EII distributions of each place emotion category under various regions with various culture differences. Regions like Oceania and South America exhibit high positive emotions, which can be attributed to their graceful natural landscapes [55,56]. In contrast, the predominance of neutral emotions in Europe and North America might be due to the fact that Europe has more human attractions, such as the British museum and Balboa Park. Africa's low negative emotions reflect their unique and more differentiated tourism cultures and life contexts. These findings suggest that different regions have unique attraction cultures, influencing how visitors experience and enjoy these destinations.

Despite the emotional differences observed among regions, radar charts show that tourist attractions across all regions elicit primarily positive and neutral emotions, with negative emotions being consistently low. This finding highlight that although the types of tourist attractions and cultural differences across regions slightly influence the emotional distribution of tourists, overall tourist satisfaction remains high globally. This is largely due to the inherent appeal of scenic and culturally rich destinations [57].

5.4. Framework Performance, Limitations, and Future Perspective

Explicitly mapping human emotions of places has long been hampered by the lack of automatic methods. This study proposed the M-GFN, a deep learning model that effectively maps place emotions in tourist attractions, which effectively suppresses the side influence of the gap between facial expressions and place emotions due to possible posed expressions in social media [58]. The high overall accuracy suggests its robust performance in extracting and analyzing tourist place emotions.

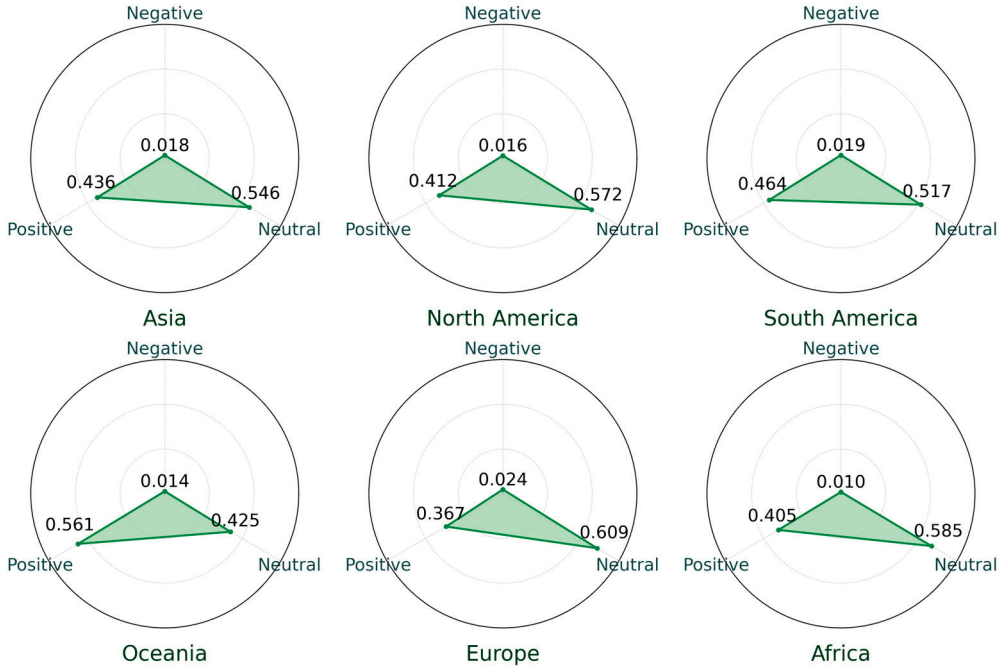


Figure 17. The EII distributions of three classes of emotions for 6 regions with various cultures.

Instead of the single smile emotion analysis, our framework used multi-view emotions to mine tourist place emotions from attraction type perspectives. Notably, more than half of the people express neutral feelings across attractions, unlike the assumption of most positive emotions in tourist places like shopping areas. There is a need to understand the detailed influences on emotions at tourist sites.

Despite its innovation, this study has limitations. Foremost, exploring the reasons behind the observed emotional differences or similarities across attractions is important. In this research, our primary goal was to develop a novel framework for tourist place emotion extraction using multi-view emotions, only briefly exploring the link between place emotions and attraction types due to the coarseness of the global dataset. In the future, we will differentiate between vacationers and regular visitors, analyzing emotional changes over time. Meanwhile, we will introduce time variation analysis and special event detection, like political situations and pandemics, to understand emotional cultural backgrounds, trends, and potential anomalies.

6. Conclusions

We presented a novel framework to extract and analyze emotions associated with various tourist attractions, leveraging multi-view emotions and attraction-specific indices from user-generated geo-tagged photos (UGPs). This is a growing form of urban big data that has permeated through urban informatics but is not yet used for such a purpose. Taking world-wide tourist attractions as diverse examples, we deployed a Multi-view Graph Fusion Network (M-GFN) to identify multi-view emotions, incorporating crowd facial emotions and implicit scene sentiments, effectively bridging the emotional gap posed by individual expressions in UGPs. Subsequently, we employ three attraction-specific indices—emotion intensity index (EII), attraction-specific emotion Index (AEI), and Pearson correlation coefficient (PCC) -to enhance the extraction and analysis of emotions at tourist locations by considering the diversity of attraction types.

The M-GFN model, which integrates meaningful environmental contexts, outperforms state-of-the-art (SOTA) methods in predicting multi-view emotions, highlighting its accuracy and establishing a solid foundation for the precise extraction of tourist place emotions. A quantitative comparative analysis conducted on the TPE dataset confirms our framework aligns with established emotional insights and outperforms another facial

expression-based framework, thereby validating its effectiveness. Furthermore, our study exploring the relationships between types and regions of tourist attractions and place emotions reveals the diverse emotional effects that different attractions exert on visitors, while also highlighting common emotional impacts, providing essential insights into the emotional dynamics at global tourist sites.

This comprehensive framework not only advances the methodology of analyzing place emotions in global tourist destinations but also has significant implications for environmental management and tourism planning. This involves utilizing emotional data to balance tourism growth with environmental conservation, ultimately enhancing the emotional experiences of visitors at tourist attractions worldwide. By applying this approach, user-generated photos can be used more widely, such as urban perception studies at different spatial scales.

Author Contributions: Data curation, Software, Visualization, Methodology, Writing—original draft, Yu Wang; Validation, Supervision, Writing—review and editing, Shunping Zhou; Writing—review and editing, Qingfeng Guan; Writing—review and editing, Fang Fang; Writing—review and editing, Ni Yang; Software, Kanglin Li; Conceptualization, Methodology, Validation, Project administration, Writing—review and editing, Yuanyuan Liu. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China grant (No. 62076227), Natural Science Foundation of Hubei Province grant (No. 2023AFB527), and Hubei Key Laboratory of Intelligent Geo-Information Processing (No. KLIGIP-2022-B10).

Data Availability Statement: The user-generated geo-tagged photos that support the experiment of this study cannot be made public due to the data-use restrictions.

Acknowledgments: We thank people who have helped shape our knowledge. Their wisdom, generosity, and patience form the bedrock of our collaborative research frameworks and programs.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ma, Z. Deep exploration of street view features for identifying urban vitality: A case study of Qingdao city. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *123*, 103476. [CrossRef]
2. Huang, Y.; Li, J.; Wu, G.; Fei, T. Quantifying the bias in place emotion extracted from photos on social networking sites: A case study on a university campus. *Cities* **2020**, *102*, 102719. [CrossRef]
3. Huai, S.; Van de Voorde, T. Which environmental features contribute to positive and negative perceptions of urban parks? A cross-cultural comparison using online reviews and Natural Language Processing methods. *Landsc. Urban Plan.* **2022**, *218*, 104307. [CrossRef]
4. Wei, H.; Ma, B.; Hauer, R.J.; Liu, C.; Chen, X.; He, X. Relationship between environmental factors and facial expressions of visitors during the urban forest experience. *Urban For. Urban Green.* **2020**, *53*, 126699. [CrossRef]
5. Zhou, H.; Wang, J.; Wilson, K. Impacts of perceived safety and beauty of park environments on time spent in parks: Examining the potential of street view imagery and phone-based GPS data. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *115*, 103078. [CrossRef]
6. Chen, X.; Li, J.; Han, W.; Liu, S. Urban Tourism Destination Image Perception Based on LDA Integrating Social Network and Emotion Analysis: The Example of Wuhan. *Sustainability* **2021**, *14*, 12. [CrossRef]
7. Liu, Y.; Wang, W.; Feng, C.; Zhang, H.; Chen, Z.; Zhan, Y. Expression snippet transformer for robust video-based facial expression recognition. *Pattern Recognit.* **2023**, *138*, 109368. [CrossRef]
8. Yin, G.; Liu, Y.; Liu, T.; Zhang, H.; Fang, F.; Tang, C.; Jiang, L. Token-disentangling Mutual Transformer for multimodal emotion recognition. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108348. [CrossRef]
9. Zhou, T.; Cai, Z.; Liu, F.; Su, J. In Pursuit of Beauty: Aesthetic-Aware and Context-Adaptive Photo Selection in Crowdsensing. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 9364–9377. [CrossRef]
10. Harb, J.G.D.; Ebeling, R.; Becker, K. A framework to analyze the emotional reactions to mass violent events on Twitter and influential factors. *Inf. Process. Manag.* **2020**, *57*, 102372. [CrossRef]
11. Ekman, P. Facial expression and emotion. *AmP* **1993**, *48*, 384. [CrossRef] [PubMed]
12. Frijda, N.H. *The Emotions*; Cambridge University Press: Cambridge, UK, 1986.
13. Kaplan, R.; Kaplan, S. *The Experience of Nature: A Psychological Perspective*; Cambridge University Press: Cambridge, UK, 1989.
14. Kang, Y.; Zeng, X.; Zhang, Z.; Wang, Y.; Fei, T. Who are happier? Spatio-temporal analysis of worldwide human emotion based on geo-crowdsourcing faces. In Proceedings of the 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS), Wuhan, China, 22–23 March 2018; pp. 1–8.

15. Ashkezari-Toussi, S.; Kamel, M.; Sadoghi-Yazdi, H. Emotional maps based on social networks data to analyze cities emotional structure and measure their emotional similarity. *Cities* **2019**, *86*, 113–124. [CrossRef]
16. Lengen, C.; Kistemann, T. Sense of place and place identity: Review of neuroscientific evidence. *Health Place* **2012**, *18*, 1162–1171. [CrossRef] [PubMed]
17. Zhu, X.; Gao, M.; Zhang, R.; Zhang, B. Quantifying emotional differences in urban green spaces extracted from photos on social networking sites: A study of 34 parks in three cities in northern China. *Urban For. Urban Green.* **2021**, *62*, 127133. [CrossRef]
18. Kosti, R.; Alvarez, J.M.; Recasens, A.; Lapedriza, A. Context Based Emotion Recognition Using EMOTIC Dataset. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2755–2766. [CrossRef] [PubMed]
19. Sisco, M.R. The effects of weather experiences on climate change attitudes and behaviors. *Curr. Opin. Environ. Sustain.* **2021**, *52*, 111–117. [CrossRef]
20. Zheng, Q.; Zhao, P.; Wang, H.; Elhanashi, A.; Saponara, S. Fine-Grained Modulation Classification Using Multi-Scale Radio Transformer with Dual-Channel Representation. *IEEE Commun. Lett.* **2022**, *26*, 1298–1302. [CrossRef]
21. Zheng, Q.; Zhao, P.; Zhang, D.; Wang, H. MR-DCAE: Manifold regularization-based deep convolutional autoencoder for unauthorized broadcasting identification. *Int. J. Intell. Syst.* **2021**, *36*, 7204–7238. [CrossRef]
22. Zhang, F.; Liu, K.; Liu, Y.; Wang, C.; Zhou, W.; Zhang, H.; Wang, L. Multitarget Domain Adaptation Building Instance Extraction of Remote Sensing Imagery with Domain-Common Approximation Learning. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–16. [CrossRef]
23. Li, Y.; Fei, T.; Huang, Y.; Li, J.; Li, X.; Zhang, F.; Kang, Y.; Wu, G. Emotional habitat: Mapping the global geographic distribution of human emotion with physical environmental factors using a species distribution model. *Int. J. Geogr. Inf. Sci.* **2021**, *35*, 227–249. [CrossRef]
24. Svoray, T.; Dorman, M.; Shahar, G.; Kloog, I. Demonstrating the effect of exposure to nature on happy facial expressions via Flickr data: Advantages of non-intrusive social network data analyses and geoinformatics methodologies. *J. Environ. Psychol.* **2018**, *58*, 93–100. [CrossRef]
25. Hernwall, P.; Siibak, A. Writing Identity: Gendered values and user content creation in SNS interaction among Estonian and Swedish tweens. *Glob. Stud. Child.* **2011**, *1*, 365–376. [CrossRef]
26. Joye, Y.; Kahn, P.H.; Hasbach, P.H. *Can Architecture Become Second Nature? An Emotion-based Approach to Nature-oriented Architecture*; MIT Press: Cambridge, MA, USA, 2012.
27. Relph, E. *Place and Placelessness*; Pion London: London, UK, 1976; Volume 67.
28. Junot, A.; Paquet, Y.; Fenouillet, F. Place attachment influence on human well-being and general pro-environmental behaviors. *J. Theor. Soc. Psychol.* **2018**, *2*, 49–57. [CrossRef]
29. Gehl, J. *Life between Buildings*; VAN Nosrand Reinhold: New York, NY, USA, 2003.
30. Moyle, B.D.; Moyle, C.-l.; Bec, A.; Scott, N. The next frontier in tourism emotion research. *Curr. Issues Tour.* **2019**, *22*, 1393–1399. [CrossRef]
31. Cheung, M.L.; Ting, H.; Cheah, J.-H.; Sharipudin, M.-N.S. Examining the role of social media-based destination brand community in evoking tourists' emotions and intention to co-create and visit. *J. Prod. Brand Manag.* **2021**, *30*, 28–43. [CrossRef]
32. Mehra, P. Unexpected surprise: Emotion analysis and aspect based sentiment analysis (ABSA) of user generated comments to study behavioral intentions of tourists. *Tour. Manag. Perspect.* **2023**, *45*, 101063. [CrossRef]
33. Jiang, M.; Li, J.; Du, Y. From on-site to memory: Study on the spatial characteristics of tourists' emotional experiences. *J. Qual. Assur. Hosp. Tour.* **2023**, *24*, 279–310. [CrossRef]
34. Guo, X.; Polania, L.; Zhu, B.; Boncelet, C.; Barner, K. Graph neural networks for image understanding based on multiple cues: Group emotion recognition and event recognition as use cases. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 2921–2930.
35. Wan, T.; Jun, H.; Zhang, H.; Pan, W.; Hua, H. Kappa coefficient: A popular measure of rater agreement. *Shanghai Arch. Psychiatry* **2015**, *27*, 62.
36. Petroman, C. Typology of tourism destinations. *Sci. Pap. Anim. Sci. Biotechnol.* **2015**, *48*, 338–342.
37. Cohen, I.; Huang, Y.; Chen, J.; Benesty, J.; Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4. [CrossRef]
38. Deng, J.; Guo, J.; Ververas, E.; Kotsia, I.; Zafeiriou, S. Retinaface: Single-shot multi-level face localisation in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 5203–5212.
39. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *99*, 2969239–2969250. [CrossRef]
40. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
41. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
42. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
43. Taecharungroj, V.; Mathayomchan, B. Analysing TripAdvisor reviews of tourist attractions in Phuket, Thailand. *Tour. Manag.* **2019**, *75*, 550–568. [CrossRef]

44. Dwyer, L.; Tomljenović, R.; Čorak, S. Evolution of destination planning and strategy. In *The rise of tourism in Croatia*; Springer: Berlin/Heidelberg, Germany, 2017.
45. Lee, J.; Kim, S.; Kim, S.; Park, J.; Sohn, K. Context-aware emotion recognition networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 10143–10152.
46. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef] [PubMed]
47. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
48. Wang, X.; Zhang, D.; Tan, H.-Z.; Lee, D.-J. A self-fusion network based on contrastive learning for group emotion recognition. *IEEE Trans. Comput. Soc. Syst.* **2022**, *10*, 458–469. [CrossRef]
49. Kang, Y.; Jia, Q.; Gao, S.; Zeng, X.; Wang, Y.; Angsuesser, S.; Liu, Y.; Ye, X.; Fei, T. Extracting human emotions at different places based on facial expressions and spatial clustering analysis. *Trans. GIS* **2019**, *23*, 450–480. [CrossRef]
50. Jones, S. The mediating effects of facial expression on spatial interference between gaze direction and gaze location. *J. Gen. Psychol.* **2015**, *142*, 106–117. [CrossRef]
51. Yüksel, A. Tourist shopping habitat: Effects on emotions, shopping value and behaviours. *Tour. Manag.* **2007**, *28*, 58–69. [CrossRef]
52. Lowenthal, D. *The Past Is a Foreign Country-Revisited*; Cambridge University Press: Cambridge, UK, 2015.
53. Kim, J.-H. The antecedents of memorable tourism experiences: The development of a scale to measure the destination attributes associated with memorable experiences. *Tour. Manag.* **2014**, *44*, 34–45. [CrossRef]
54. Zatori, A.; Smith, M.K.; Puczko, L. Experience-involvement, memorability and authenticity: The service provider's effect on tourist experience. *Tour. Manag.* **2018**, *67*, 111–126. [CrossRef]
55. D'Arcy, P. Oceania: The Environmental History of One-Third of the Globe. *A Companion Glob. Environ. Hist.* **2012**, 196–221.
56. Vincent, F. *Around and about South America*; D. Appleton & Company: New York, NY, USA, 1890.
57. Vittersø, J.; Vorkinn, M.; Vistad, O.I.; Vaagland, J. Tourist experiences and attractions. *Ann. Tour. Res.* **2000**, *27*, 432–450. [CrossRef]
58. Singh, V.K.; Atrey, A.; Hegde, S. Do individuals smile more in diverse social company? Studying smiles and diversity via social media photos. In Proceedings of the 25th ACM international conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 1818–1827.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

Extracting Geoscientific Dataset Names from the Literature Based on the Hierarchical Temporal Memory Model

Kai Wu [1,2], Zugang Chen [1,*], Xinqian Wu [2], Guoqing Li [1], Jing Li [1], Shaohua Wang [1], Haodong Wang [3] and Hang Feng [3]

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; wk_wk@stu.haust.edu.cn (K.W.); ligq@aircas.ac.cn (G.L.); lijing6@aircas.ac.cn (J.L.); wangshaohua@aircas.ac.cn (S.W.)
[2] School of Mathematics and Statistics, Henan University of Science and Technology, Luoyang 471023, China; wxq1001@haust.edu.cn
[3] School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China; zzuwhd@gs.zzu.edu.cn (H.W.); feng_hang_fh@gs.zzu.edu.cn (H.F.)
* Correspondence: chenzg@aircas.ac.cn

Abstract: Extracting geoscientific dataset names from the literature is crucial for building a literature–data association network, which can help readers access the data quickly through the Internet. However, the existing named-entity extraction methods have low accuracy in extracting geoscientific dataset names from unstructured text because geoscientific dataset names are a complex combination of multiple elements, such as geospatial coverage, temporal coverage, scale or resolution, theme content, and version. This paper proposes a new method based on the hierarchical temporal memory (HTM) model, a brain-inspired neural network with superior performance in high-level cognitive tasks, to accurately extract geoscientific dataset names from unstructured text. First, a word-encoding method based on the Unicode values of characters for the HTM model was proposed. Then, over 12,000 dataset names were collected from geoscience data-sharing websites and encoded into binary vectors to train the HTM model. We conceived a new classifier scheme for the HTM model that decodes the predictive vector for the encoder of the next word so that the similarity of the encoders of the predictive next word and the real next word can be computed. If the similarity is greater than a specified threshold, the real next word can be regarded as part of the name, and a successive word set forms the full geoscientific dataset name. We used the trained HTM model to extract geoscientific dataset names from 100 papers. Our method achieved an F1-score of 0.727, outperforming the GPT-4- and Claude-3-based few-shot learning (FSL) method, with F1-scores of 0.698 and 0.72, respectively.

Keywords: geoscientific dataset; named-entity recognition; hierarchical temporal memory; word encoding

1. Introduction

Geoscientific datasets include data describing the state, properties, and distribution characteristics of phenomena or entities in specific layers or geographic locations on Earth [1–3]. These datasets are widely used in atmospheric, oceanic, geological, terrestrial surface, and solar–terrestrial space science research [4]. The scientific and technical literature [5] is essential for the acquisition of knowledge by researchers as an important source of recorded scientific discoveries and innovations. With the rise and rapid development of open science [6], an increasing number of geoscientific datasets and papers are being published and shared on the Internet. In the field of geoscience, scientific research is increasingly dependent on geoscientific datasets in the context of a data-intensive scientific research paradigm [7]; thus, the literature contains extensive geoscientific dataset names.

However, in most cases, geoscientific datasets and papers are scattered in different corners of the Internet, and correlations between them have not been established.

Extracting the names of geoscientific datasets from the literature can clarify which datasets are used in research. It can further help us to discover these datasets through the Internet and establish correlations between studies and datasets. Consequently, this can help readers quickly access research data when they read the literature online and rapidly reproduce studies [8,9]. Furthermore, extracting geoscientific dataset names can assist data-publishing journals, such as *Earth System Science Data*, in tracking the citations of their published datasets [10,11].

The name of a geoscientific dataset generally contains elements such as geospatial coverage, temporal coverage, scale or resolution, theme content, and version [12,13]. For example, "San Francisco 5-m resolution land use data v1.0 for the years 2015–2020" represents a typical name for a geoscientific dataset. In this instance, "San Francisco" denotes the geospatial coverage of the dataset, "5-m resolution" indicates the spatial resolution of the dataset, "land use data" describes the content of the dataset, "v1.0" signifies the dataset's version, and "years 2015–2020" indicates the temporal coverage of the dataset. Extracting dataset names from the literature is also the task of domain-specific named-entity recognition methods [14,15]. Although existing methods have made some progress in named-entity recognition (NER), they still face many challenges in extracting geoscientific dataset names. Because the names of geoscientific datasets contain many elements and there can be many combinations of elements, it is hard for existing methods to understand the rules in the names. Therefore, existing methods often can only extract some elements of the name, not the complete name, which results in low extraction accuracy.

To address these problems, this study proposes a method for extracting geoscientific dataset names based on the hierarchical temporal memory (HTM) model. The HTM model is a brain-inspired artificial neural network [16,17] that has excellent abilities in high-level cognitive activities, such as language understanding, spatial cognition, and navigation [18]. The elements of dataset names are encoded into binary vectors by our proposed new encoding method. The sequence of binary vectors of a dataset name is input to train the HTM model and make it understand the naming rule. We then use the trained HTM model to recognize the names of geoscientific datasets from text in the literature. Compared with existing methods such as GPT-4 and Claude-3, the results show that our method has a higher F1-score.

The main contributions of the paper are as follows:

(1) An artificial neural network method developed specifically for extracting the names of geoscientific datasets is proposed. Compared with the GPT-4-based few-shot learning (FSL) method, this method has a higher F1-score.
(2) A new word-encoding method for the HTM model is proposed. This method uses the Unicode values of characters to determine their relative positions in the semantic vector and employs a numerical mapping approach to reduce the encoding length. Compared to the semantic folding method, this approach demonstrates higher accuracy in encoding Chinese characters.
(3) A new decoding structure for the HTM model is proposed. This decoding structure uses a BP neural network to decode the prediction vector for the encoding of the next word. By calculating the similarity of the encoding of the predicted word and the actual next word, the name of the geoscientific dataset is extracted word by word.

The remainder of the paper is organized as follows. Section 2 describes the state of the art. Section 3 presents the general idea and methodology. Section 4 mainly describes the experimental design for extracting geoscientific dataset names using the proposed method and two large language models—GPT-4 and Claude-3. Section 5 evaluates and compares our method with existing methods. The last section concludes the research and discusses the future research direction.

2. Related Work

We propose a geoscientific dataset name-extraction method based on the HTM model. Therefore, the reviewed literature includes geoscientific dataset name-extraction methods and HTM-based algorithms.

2.1. Progress in Geoscientific Dataset Name Extraction

Different from traditional personal or place-name extraction, most existing named-entity recognition (NER) methods are not suitable for geoscientific datasets because these datasets have complex and compound elements. Even so, some researchers have proposed a few geoscientific dataset name or related name-entity extraction methods. These methods can be classified into three categories.

The first category is rule-based methods. A rule-based method requires the construction of a rule database. For example, Cao et al. [19] used regular expressions to create dataset name-extraction rules and successfully extracted dataset names from the geoscience literature with 62% accuracy. Afzal et al. [20] proposed a rule-based citation mining technique. This technique detected data such as author, title, and conference location from documents and extracted the relevant titles from the computer science literature database of the Digital Bibliography Library Project (DBLP). Rule-based methods require the manual creation of explicit rules and are time-consuming [21,22]. Moreover, for complex named entities, it is difficult to establish a complete rule database, so the accuracy is not high.

The second category is traditional machine learning methods. Machine learning methods generally use the inside, outside, beginning (IOB) annotation system to manually annotate the corpus and then use machine learning models for training and prediction [23]. Commonly used machine learning methods for named-entity recognition include the hidden Markov model (HMM) [24], support vector machine (SVM) [25], and conditional random fields (CRF) [26]. Han et al. [27] proposed an SVM-based method for extracting metadata from a structured corpus that outperformed other machine learning methods. Although HMM, SVM, and CRF can theoretically be used for geoscientific data name extraction, in reality, to the best of our knowledge, no studies have used these methods to directly extract geoscientific data names.

The third category is deep learning-based methods. In recent years, deep learning methods [28] such as the large language model (LLM) have achieved remarkable results in the field of natural language processing. Deep learning-based methods can automatically learn features from data and effectively process large text data. Commonly used deep learning methods for named-entity recognition include convolutional neural network (CNN) [29], bidirectional long short-term memory (Bi-LSTM) [30], transformer models [31], and GPT-4 [32]. Yao et al. [33] extracted dataset names and methods based on the Bi-LSTM model from the papers of PAKDD conferences (2009–2019). Kumar et al. [34] tested dataset name extraction based on the bidirectional encoder representations from transformers (BERT) model on research papers in a popular social science corpus and achieved an F1-score of 56.2%. Younes and Scherp [35] proposed one-step and two-step methods to extract unknown dataset names from scientific papers. The one-step method had higher accuracy, while the two-step method could extract more potential datasets. Heddes et al. [15] annotated 6000 sentences from AI conferences and used the SciBERT method, a pretrained LLM for scientific text [36], to automatically detect dataset names from scientific articles. Geogalactica [37], which was trained on a geoscience-related text corpus, also demonstrated advanced performance in various NLP tasks, including the extraction of key information in geoscience.

From the research above, it can be found that little research has been conducted on directly extracting geoscientific dataset names from unstructured text, such as the geoscientific literature. The existing methods either have low accuracy or are not specifically designed for geoscientific datasets. LLMs [37,38], which are reported to have impressive performance in NLP tasks including NER, may not perform as well in the extraction of geoscientific datasets, as geoscientific datasets involve a lot of specialized knowledge and have complex combination patterns. Thus, high-precision methods for extracting geoscientific data names are urgently

needed to associate the data with scientific studies, to measure the usage of scientific data in the scientific literature, and to reproduce and reuse the research results of the studies.

2.2. Progress in HTM Application

Time is an important aspect of vocabulary embedding and neural network training. Time not only influences the semantics of text but also reveals the dynamic changes in text. Therefore, many time-aware text embedding methods have been proposed, such as time-aware text embedding approach to generate subgraphs (TEAGS) [39], short-text author linking through multi-aspect temporal-textual embedding (SoulMate) [40], the multi-aspect time-related influence (MATI) model [41], which integrates multiple time features and combines the temporal subset property (TSP) [42], and the value-wise ConvNet for transformer models [43]. These methods highlight the importance of the temporal dimension in text embedding and demonstrate significant advantages in time-sensitive recommendation tasks.

Realizing the importance of the temporal dimension in data processing, George and Hawkins [16,17] proposed the hierarchical temporal memory (HTM) model, which was designed to simulate the structure and function of the human brain and had excellent performance in time series data processing. Compared to the traditional neurons of an artificial neural network, HTM has a great number of dendritic connections, and the output of HTM neurons depends not only on feedforward inputs (proximal connections to the soma) but also on lateral connections between the neurons. Thus, it can perform high-level cognitive activities, such as language analysis, spatial cognition, etc. It is widely applied in anomaly detection [44,45], image processing [46], natural language processing [47], and other areas. For instance, Afaf et al. [48] applied the HTM algorithm to traffic congestion detection, demonstrating a 7.4% improvement in detection accuracy compared to state-of-the-art techniques, with an average F-score of 98.83%. Szoplák et al. [49] tested the HTM algorithm on the PAN plagiarism corpus, achieving 70.15% accuracy in detecting anomalous texts. Hamid et al. [50] proposed a text anomaly detection framework based on the improved semantic folding theory (SFT), achieving 96% accuracy on the Yelp dataset. HTM has unique advantages in handling continuous streaming data and capturing temporal patterns, although it may require more training when adapting to new data [51,52]. The initial version of HTM has now been upgraded to the third generation [53,54].

In our research, we relied on the excellent ability of HTM to process time-series data to realize the extraction of geoscientific dataset names from text. This is a newly proposed method that encodes words to binary vectors and uses a large amount of geoscientific dataset names for training the HTM model, which allows the model to master the naming rules of geoscientific datasets and the combination rules of different name elements. Then, we used the trained HTM model to extract geoscientific dataset names from the literature. Finally, we compared our method with existing methods in relation to extraction accuracy.

3. Methodology

3.1. General Idea

The input of the HTM model is binary vectors. In order to deal with the name text of geoscientific datasets, we propose a new word-encoding method to convert words to binary vectors. Then, using the HTM model, we can extract geoscientific dataset names based on the word-encoding method. We collected geoscientific dataset names from geoscientific data-sharing websites as a corpus or sample to train the created HTM model. We segmented the geoscientific dataset names into a sequence of words by using the NLPIR tool [55]. Each word represented a feature of the geoscientific dataset and was encoded and input into the HTM model, enabling it to learn the naming rules and combination patterns of various features. A trained HTM neural network was obtained by the offline section. Then, we used the trained HTM model to extract geoscientific names by the online section as follows: For arbitrary unstructured text from geoscientific papers, we segmented a sentence from the text into word sequences using the NLPIR tool [55]. The word was encoded and input to the trained HTM model, and the HTM model generated a predictive vector. In this

research, we decoded the predictive vector for the next word's encoding vector by using a BP neural network as the classifier for the HTM model so that we could use the similarity of encoding of the predictive next word and real next word to determine whether the real word was part of a geoscientific dataset name. Finally, the combination of consecutive multiple words was regarded as a geoscientific dataset name, thereby achieving the goal of automatically extracting a geoscientific dataset name from unstructured text.

Finally, we evaluated our method by comparing it with zero-shot learning (ZSL) and few-shot learning (FSL) methods, which were based on GPT-4 and Claude-3, and we evaluated the precision and recall of the proposed method in the task of extracting geoscientific dataset names. The general idea is shown in Figure 1.

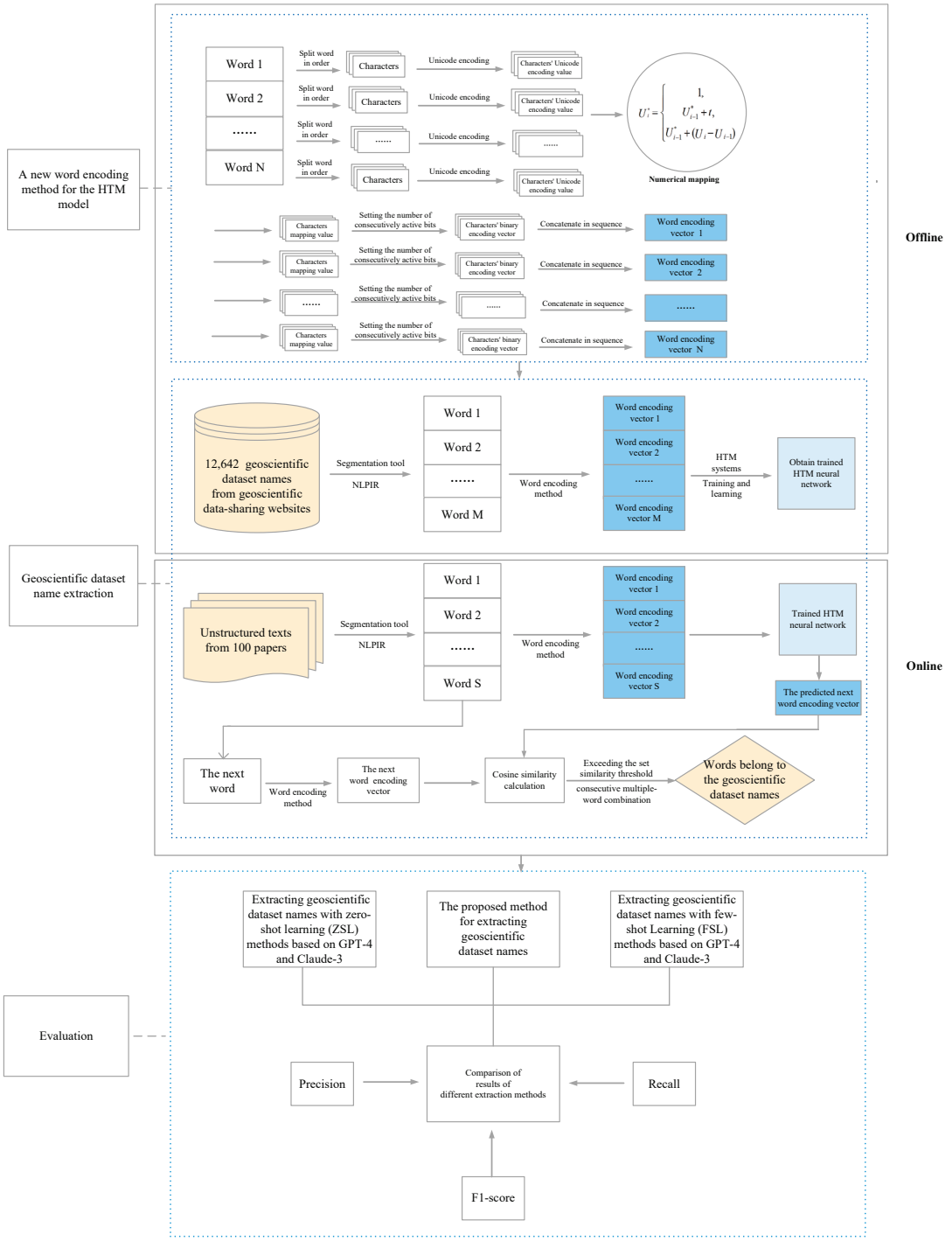


Figure 1. General idea of the paper.

3.2. HTM Model

Hierarchical temporal memory (HTM) is a biomimetic machine intelligence algorithm [52] that was first proposed by George and Hawkins [16,17]. HTM is inspired by the structure of the cerebral cortex of the human brain and has more fidelity than traditional artificial neural networks [56].

A primary feature of HTM is its hierarchical structure; it is composed of several layers with distinct regions (Figure 2A). The spatial pooler (SP) and the temporal memory (TM) are core components of HTM (Figure 2B). The SP converts all input patterns into sparse distributed representations (SDRs), which are used as the input of the TM. The TM learns the temporal sequences of the SDRs and can predict the next input of the HTM. HTM neurons (Figure 2C) are similar to biological neurons, with thousands of synapses on activated dendrites, and the model learns by simulating the growth of new synapses and the decay of inactive synapses. The proximal and distal dendritic segments of the HTM neurons have different functions [57]. Patterns detected on the proximal dendrites lead to action potentials (i.e., they are activated) (Figure 2D) and define the classic receptive field of neurons. Patterns recognized by the distal synapses of neurons act as predictors by depolarizing the cell without directly causing an action potential [58]. The output of HTM neurons depends not only on feedforward inputs (Figure 2E) but also on lateral connections between neurons. The structure of HTM is shown in Figure 2.

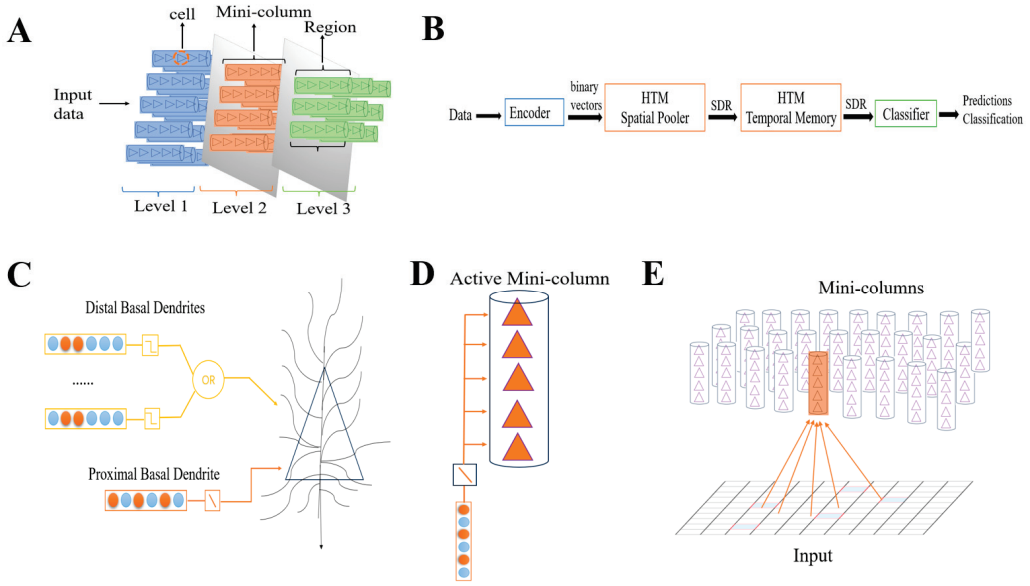


Figure 2. HTM structure [57,58]. (**A**) HTM has a three-level hierarchy. The smallest unit is an HTM cell. In each layer, there are a large number of cells, multiple cells form mini-columns, and multiple mini-columns form regions. (**B**) The end-to-end HTM system includes an encoder, HTM SP, HTM TM, and a classifier. (**C**) An HTM neuron has one proximal dendrite and several distal dendrites, and dendrites have different functions. Proximal dendrites receive feedforward inputs, while distal dendrites receive contextual information from nearby cells in the layer. (**D**) All cells in the same mini-column share the same synapses that receive feedforward inputs, which means they receive the same information. (**E**) Each layer of the HTM model consists of several mini-columns of cells that can read and form synaptic connections with input data.

3.2.1. Sparse Distributed Representations

A sparse distributed representation (SDR) is a sparse, high-dimensional binary vector consisting of a large number of zeros and ones [59]. At every point in time, only a few encoding sites are activated and have a value of 1, which corresponds to active neurons in the mammalian brain, while the rest of the coding sites have a value of 0. Although a single

encoding site does not represent valid information, the overall combination of encoding sites has a certain semantic meaning.

3.2.2. A New Word-Encoding Method for the HTM Model

The HTM encoder encodes the input data into a binary vector that can be processed by the SP layer. The purpose of the encoding process is to determine which output bits are zeros and which are ones, to capture the semantic features of the data. The encoder should conform to four principles: the encoder of input data has the same dimension, the same data are encoded identically, similar data are encoded similarly, and the output sparsity of different data remains essentially the same [60].

A statistical analysis of the Unicode distribution of Chinese characters shows that the Unicode values of semantically similar characters are arranged in nearby locations [61,62]. This study proposes a new word-encoding method for HTM based on the Unicode and arranging law of characters. First, the Unicode values of the 3596 commonly used Chinese characters are obtained; the minimum Unicode value is 40, and the maximum is 65,311. To decrease the dimension of the encoder, the Unicode values of the 3596 characters are mapped to new values with a more compact distribution. The following method is used:

$$U_i^* = \begin{cases} 1, & U_1 \\ U_{i-1}^* + t, & |U_i - U_{i-1}| > t, \\ U_{i-1}^* + (U_i - U_{i-1}), & 0 < |U_i - U_{i-1}| \le t \end{cases} \quad i \text{ is a positive integer.} \tag{1}$$

where U_1, U_2, …, U_{i-1}, U_i, …, U_n represents the Unicode values for the 3596 commonly used Chinese characters, letters, symbols, and numerals, sorted in ascending order. U_i^* is the conversion value, and t is a threshold that controls the total number of encoding bits and the degree of semantic generalization.

Secondly, we use the NLPIR tool to segment the names of geoscientific datasets into word sets (the maximum number of characters in the word is set) [55]. For every word, the Unicode value of each character contained in it is obtained and converted by using Formula (1). The following steps are used to encode words: (1) For all U_i^*, the minimum U_i^* is U_1^*, the maximum is denoted as $U_{\max}^*$, and the range is $[U_1^*, U_{\max}^*]$. (2) We create $U_{\max}^* - U_1^*$ buckets to split the character encoder values. (3) We choose the number of consecutive active bits w in each representation and compute the total number of bits n of the encoded character: $n = U_{\max}^* + w - 1$. For each character and its U_i^*, the encoded representation is created by setting n unset bits with w consecutive active bits starting at index U_i^*. (4) All encodings of characters in the word are combined in order. Words that do not reach the maximum length are encoded as empty with zeros to ensure that all words have the same number of bits in their encoding. An example of encoding the names of geoscientific datasets is shown in Figure 3.

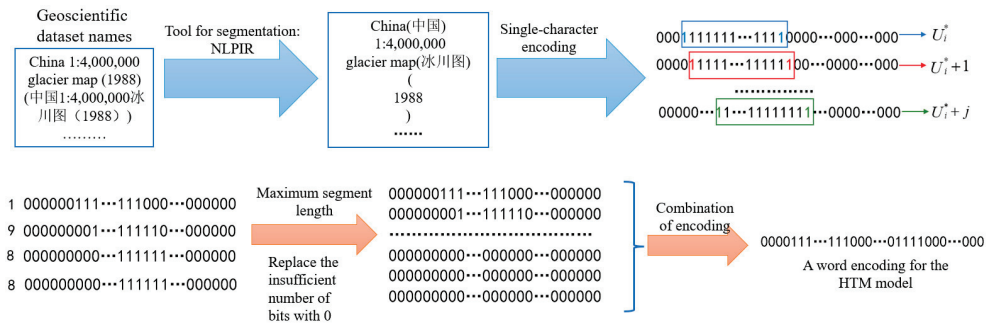


Figure 3. Example of encoding words in names of a geoscientific dataset using our method.

3.2.3. HTM Spatial Pooler

The main purposes of the spatial pooler are, first, to transform input vectors into sparse distributed representations, and second, to learn to better detect repeated input patterns. The spatial pooler algorithm is shown in Figure 4.

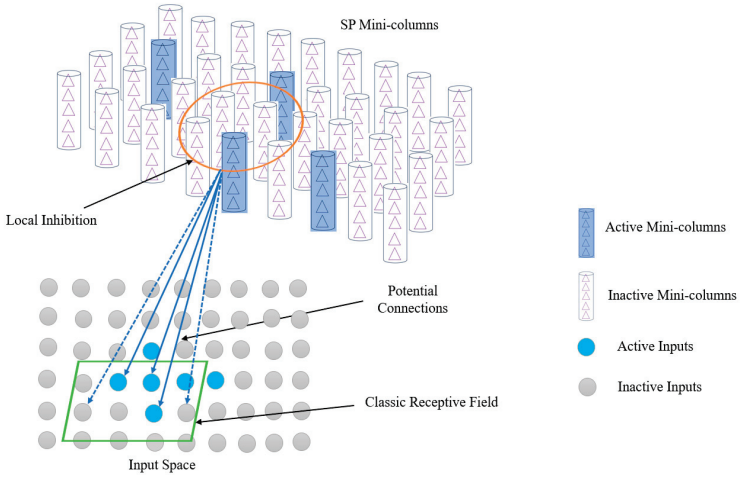


Figure 4. Structure of HTM spatial pooler [58].

The spatial pooler (SP) algorithm consists of three steps: the formation of classic receptive field (for example, the dashed green rectangle in Figure 4), the activation of mini-columns, and the learning of proximal synapse permanence. The classic receptive field formation stage determines the key parameters of the HTM model and establishes potential connections between the input data and the synapses of the proximal dendrites of mini-column k. The potential connection between mini-column k and the input data can be denoted as

$$P_k = \{i | I(x_i; x_k^c, \lambda) \; and \; (\alpha_{ki} \sim U(0,1))\}, \tag{2}$$

where x_i denotes the i-th input neuron (data), x_k^c is the central neuron of the classic receptive field of mini-column k, and λ is the length of the classic receptive field. α_{ki} is a random number chosen from a uniform distribution $U(0,1)$ and denotes the synaptic permanence between the k-th mini-column and input neuron x_i. For potential connections to become synaptic connections, the permanence of the potential synapses must exceed a given threshold η_c. $I(x_i; x_k^c, \lambda)$ is an indicator function: if and only if x_i is located in the classic receptive field of mini-column k, then the input neuron x_i is selected as the potential input of the mini-column.

Given an input pattern x, the activation of the SP mini-columns is determined based on the computation of the feedforward input, which is called overlap value O_k:

$$O_k = \beta_k \sum_i B_{ki} x_i, \tag{3}$$

$$B_{ki} = \begin{cases} 1, & \alpha_{ki} > \eta_c \\ 0, & otherwise \end{cases}, \tag{4}$$

where β_k is a positive boost factor, which controls the excitability of the SP mini-columns and is adjusted for learning during the SP training process, and B_{ki} is a binary indicator matrix that represents connected synapses, with $B_{ki} = 1$ when α_{ki} is greater than η_c; otherwise, $B_{ki} = 0$. Suppose the connection threshold for synaptic permanence η_c (values in [0, 1]) is set to be 0.7, such that initially 70% of the potential synapses are connected. The value of η_c determines the threshold for whether a synapse is considered to be connected to a neuron. If the permanence value for a synapse is greater than η_c, it is considered to be connected. According to the local inhibition mechanism, the k-th SP mini-column needs to

satisfy two conditions to be activated: first, the overlap value O_k of the k-th mini-column must be at the top n $(n < m)$ of its m neighboring mini-columns, and second, its input overlap needs to be greater than the given activation threshold θ. To ensure that a certain number of mini-columns are activated, activation threshold θ is usually set to a small positive number. Let C_k denote the activation state of the k-th SP mini-column; that is:

$$O_{Nk} = \{O_k | k \in N_k\}, \tag{5}$$

$$C_k = \begin{cases} 1, & O_k \geq prctile(O_{Nk}, 1-d) \; and \; O_k \geq \theta \\ 0, & otherwise \end{cases}. \tag{6}$$

where N_k denotes the set of neighboring mini-columns and O_{Nk} denotes the overlap value of all neighboring mini-columns of the k-th SP mini-column; $prctile(\cdot)$ denotes the percentile function; and d denotes the activation density of the target mini-column.

In the proximal synaptic permanence learning stage, the synaptic permanence values of the activation mini-columns are updated based on Hebb's learning rule. This rule means that synapse permanence is strengthened for active input connectivity bits and weakened for inactive input connectivity bits, and the synapse permanence value is limited to between zero and one.

3.2.4. HTM Temporal Memory

Temporal memory [63,64] is a predictive mechanism based on time sequence data that uses synaptic connections between cells to construct temporal associations of things with different features. Temporal memory has two main purposes: the first is to transform the SDRs of the output of the SP process into a representation that captures the temporal background of the current input, and the second is to predict future inputs based on previous sequences.

The temporal memory (TM) algorithm consists of three parts (Figure 5): determining the activated cells in the mini-column, learning the permanence of distal synapses, and obtaining the predicted cells. First, after the SP process, partially activated mini-columns are generated, and the TM process will activate a few cells in the active mini-columns based on historical information. The status of activated cells is calculated as follows:

$$\beta_{kl}^t = \begin{cases} 1, & if \; k \in c_cols^t \; and \; \mathbb{C}_{kl}^{t-1} = 1 \\ 1, & if \; k \in c_cols^t \; and \; \sum_l \mathbb{C}_{kl}^{t-1} = 0 \\ 0, & otherwise \end{cases}, \tag{7}$$

where c_cols^t denotes the mini-column activated at moment t, $\mathbb{C}_{kl}^{t-1}$ denotes the predicted state of the l-th cell of the k-th mini-column at moment $t-1$, and β_{kl}^t denotes the activation state of the l-th cell of the k-th mini-column at moment t. If a cell in the activation column at a previous moment is in a predicted state, then that cell will be activated. If no cell in the activation column is in a predicted state, then all cells in the mini-column will be activated. The TM algorithm differs from the SP algorithm in that it learns sequence correlations by activating cells in the active mini-column and represents different contexts.

Like the SP process, the TM process adjusts the permanence of distal synapses using Hebb's learning rule. Dendritic segments on learning cells connected to active cells from the previous period are reinforced, while dendritic segments on learning cells connected to inactive cells are punished. The reinforcement of dendritic segments increases the permanence value of active synapses by a larger-value τ^+ while decreasing the permanence value of inactive synapses by a smaller-value τ^-.

$$\Delta S_{kl}^d = \tau^+ \hat{S}_{kl}^d \circ A^{t-1} - \tau^- \hat{S}_{kl}^d \circ (1 - A^{t-1}), \tag{8}$$

$$\hat{S}_{kl}^d = \begin{cases} 1, & if \; S_{kl}^d > 0 \\ 0, & otherwise \end{cases}, \tag{9}$$

Suppose that the number of mini-columns in an HTM neural network is N and each column has M cells. S_{kl}^{d} is an $M \times N$ matrix denoting the persistence of the dendritic segment of the l-th cell of the k-th mini-column, A^{t} is an activated status matrix of $M \times N$ at moment t, and β_{kl}^{t} denotes the activation state of the l-th cell of the k-th mini-column. $\circ$ denotes the multiplication of matrix elements at corresponding positions, and $\hat{S}_{kl}^{d}$ denotes the matrix containing the positive terms in S_{kl}^{d}.

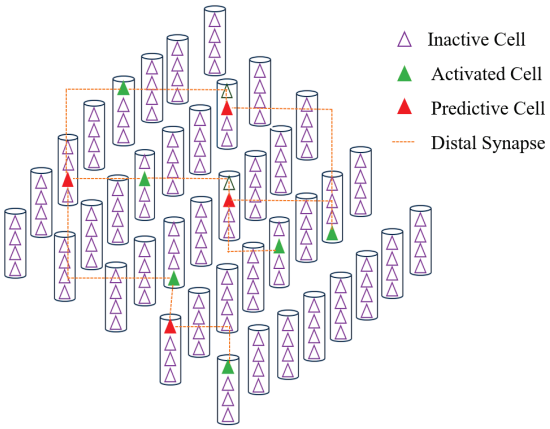


Figure 5. Structure of HTM temporal memory [63,64]. Cells in the TM process can exist in three states: inactive, active, or predictive. When a cell does not receive any feedforward input, it is in an inactive state (purple triangle), and when it receives feedforward input, it is in an active state (green triangle). Sufficient lateral activity in a contextual dendrite leads to a predictive state (red triangle).

Finally, the TM process will make predictions for the next time of input based on the number of activated cells. Whether a distal dendritic segment is active or not depends on the number of active synapses connected to active cells, and if this number exceeds a threshold ζ, the dendritic segment is activated. The dendritic segment activation threshold ζ indicates how many cells need to be activated within a time step to activate the entire mini-column, and its value is correlated with the number of cells contained in the mini-column. The value of ζ cannot exceed the number of cells contained in the mini-column. Therefore, the predicted state of the cell at moment t is:

$$\mathbb{C}_{kl}^{t} = \begin{cases} 1, & if\ \exists_d \| DS_{kl}^{d} \circ A^{t} \|_1 > \zeta \\ 0, & otherwise \end{cases}. \tag{10}$$

where DS_{kl}^{d} is the d-th distal dendritic segment of the l-th cell of the k-th cell column, and A^{t} is a 0–1 matrix with size $M \times N$.

3.2.5. Classifier

We can use common classifier algorithms such as support vector machine and naive Bayes for HTM models [65]. In this paper, text data are encoded and input into the HTM model. For the SDRs generated by the SP or TM process, the classifier can transform them into human-readable data, which is similar to the decoding process. In this paper, the BP algorithm [66,67] is used as a classifier, i.e., a decoder.

In contrast to the traditional HTM classifier, which directly transfers the prediction encoder of TM to data humans can understand, in this research, we use BP to transfer the prediction encoder of TM to the next word's encoder in the training stage. In the application stage of the trained HTM, when we encode the former word and input the encoder to the HTM, the next word's encoder is predicted. The word vector similarity between the encoders of the predictive word and real next word is computed. If the similarity is greater than a set threshold, the concatenation of the two words is deemed to conform to the naming pattern of a geoscientific dataset.

4. Experimental Section

In this section, we create a corpus (sample) and use the word-encoding method to train the created HTM model, enabling the HTM model to learn the naming rules and understand the combination patterns of different elements of the geoscientific dataset.

4.1. Training Corpus

To make the HTM model learn the naming rules and understand the combination patterns of different elements of the geoscientific dataset, we collected various geoscientific dataset names as a corpus or sample for training. The National Earth Observation Data Center (NODA) is the biggest Earth observation data center in China. About 8000 Earth observation datasets are shared in its data platform, ChinaGEOSS (https://www.chinageoss.cn/datasharing (accessed on 12 January 2024)). The National Earth System Science Data Center (NESSDC) is a key repository for Earth science data, covering a variety of data including atmospheric, oceanic, terrestrial, and ecological sciences. It aims to support Earth system research by providing a centralized platform for collecting, archiving, and disseminating data. NESSDC has shared more than 30,000 geoscientific datasets (https://www.geodata.cn/ (accessed on 12 January 2024)). We randomly selected about one-third (12,642) of the dataset names from the two national platforms to make sure that the sample has wide coverage. The NLPIR [55] was used to segment the 12,642 dataset names into words. The number of characters in a word was limited to 6. Starting and ending labels were added to all segmented names.

4.2. Creating and Training the HTM Model

We conducted experiments using the open-source HTM framework on the Numenta Platform for Intelligent Computing (NuPIC) (https://github.com/numenta/htm.java (accessed on 15 January 2024)). First, we created an HTM model. Then, we encoded all 12,642 geoscientific dataset names using the method described in Section 3.2.2, and the encodings of every word in every dataset name were input to the HTM in sequence. We trained the model on a computer with two 2.1 GHz Intel processors, 512 GB memory, 8 NVIDIA GeForce RTX 3080, and Windows 10 Professional. To obtain an HTM model with optimal performance, we tested the relationship between model size, training iteration times, and prediction accuracy. We set the number of columns of the HTM to 1000, 2000, 3000, 4000, 5000, and 6000, and set the cycles of the HTM to 50, 100, 150, 200, 250, and 300 times the number of geoscientific dataset names in the sample. We used the following formula to indicate the accuracy of training [64]:

$$accuracy = avg(\frac{|SP_words|}{|All_words|}) \tag{11}$$

where $|SP_words|$ is the number of successfully predicted words for a geoscientific dataset name, $|All_words|$ is the number of total words in the geoscientific dataset name, and $avg()$ is the average accuracy for 12,642 sample names. The prediction accuracy under different iterations and model sizes is shown in Figure 6.

As shown in Figure 6, as the model size increases, the training performance of the HTM model generally improves. When the number of columns of the HTM model is 5000, we can obtain the performance for different training iteration times. With increased training iteration times, the performance of the HTM model also generally improves; the optimal number of training iteration times is 250, and more training times will result in overfitting. The parameters of the HTM model are shown in Table 1 (the meanings of the parameters are given in detail in Appendix A). After the training, we obtained an HTM model that had learned the naming rule and combination patterns of different elements of the geoscientific dataset.

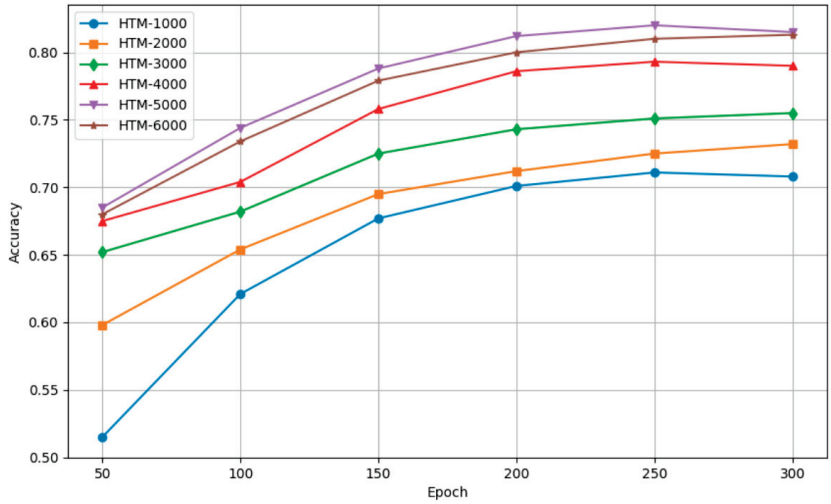


Figure 6. Prediction accuracy with different model sizes and training iteration times.

Table 1. Parameters of trained HTM model.

Parameter Name	Value	Parameter Name	Value
Number of columns	5000	Number of cells per column	6
Input dimensions	22,476	Potential radius	6
Number of active columns	48	Connected threshold for synaptic permanence	0.7
Initial synaptic permanence	0.1	Dendritic segment activation threshold	4
Synaptic permanence increment	0.1	Synaptic permanence decrement	0.1

5. Evaluation

In this research, the word-encoding method is the basis of the experiment. Whether or not the binary vector can represent the semantic features of the word has an important influence on the accuracy of extracting geoscientific dataset names. Therefore, we firstly evaluated the precision of the word-encoding method. Then we used the trained HTM model to extract geoscientific dataset names and evaluated its accuracy by comparing it with existing methods.

5.1. Evaluation of Word-Encoding Method Precision

To evaluate the precision of the proposed word-encoding method, we compared it with the semantic folding method [68], which is the main text encoding method used for the HTM model. The key to the quality of encoding methods is in their ability to convey the semantics of words, so we can use the encoding binary vectors' cosine similarity to indicate the semantic similarity of word pairs. Because we extracted names in the geoscientific domain, we select the Geo-Terminology Relatedness Dataset (GTRD) [69] which contains 66 pairs of geographic terms and corresponding semantic similarities, as the benchmark.

First, we encoded the words in the GTRD using our method and the semantic folding method. Then, we calculated the cosine similarity of the encoders for the word pairs in GTRD separately. The result is shown in Table 2. Subsequently, we calculated the Pearson correlation coefficient between the cosine similarity and the standard similarity of word pairs in the benchmark dataset. Finally, to determine whether the correlation was statistically significant, we applied the p-value test [70] to the Pearson correlation coefficient. The result is shown in Table 3.

Table 2. Cosine similarity of two encoding methods for word pairs in GTRD.

ID	Word One	Word Two	Standard Similarity	Semantic Folding Similarity	Our Encoding Method Similarity
1	waterway transportation	waterway transportation	1	1	1
2	oasis city	oasis city	1	1	1
3	port city	estuary city	0.91	0.62	0.67
4	tropical rainforest climate	equatorial rain climate	0.87	0.53	0.33
5	city	cities and towns	0.86	0.58	0.50
6	transportation	communication and transportation	0.83	1	0.71
7	nearshore environment	coastal environment	0.78	0.88	0.75
8	nearshore environment	sublittoral environment	0.78	0.96	0.50
9	plateau permafrost	frozen ground	0.78	0.22	0.58
10	cold wave	cold air mass	0.77	0.64	0
11	iron and steel industry	metallurgical industry	0.73	0.56	0.72
12	geographical environment	environment	0.71	1	0.71
13	highway transport	transport	0.71	1	0.71
14	semi-arid climate	steppe climate	0.71	0.67	0.45
15	climate	weather	0.69	0.23	0
16	milk industry	food industry	0.68	0.7	0.75
17	cultural landscape	landscape	0.67	1	0.71
18	processing industry	light industry	0.66	0.66	0.58
19	cold wave	disastrous weather	0.65	0.28	0.03
20	coal industry	heavy industry	0.63	0.61	0.58
21	farming industry	industry	0.61	1	0.53
22	gray desert soil	brown desert soil	0.60	0.82	0.67
23	marine environment	geographical environment	0.60	0.61	0.50
24	eco-environment	water environment	0.59	0.61	0.58
25	tropical soil	subtropical soil	0.57	0.84	0.89
	...	...	...	...	...
64	desert climate	internal water transport	0.03	0.26	0.01
65	polar climate	mining industry	0.02	0.25	0
66	desert climate	labor-intensive industry	0	0.2	0

Table 3. Pearson correlation coefficient and *p*-value test between proposed encoding method and semantic folding method on GTRD.

Statistical Indicator	Proposed Encoding Method	Semantic Folding Method
Pearson correlation coefficient	0.69	0.62
p-value	2.15×10^{-10}	3.61×10^{-8}

From Table 3, it is evident that on the benchmark GTRD dataset, our proposed encoding method outperforms the semantic folding approach in terms of accuracy.

5.2. Evaluation of Geoscientific Dataset Name-Extraction Accuracy

To comprehensively evaluate the ability of HTM model to extract geoscientific dataset names, we compared it with two large language models, GPT-4 and Claude-3, which have excellent performance in named-entity recognition [71,72]. First, we created a baseline based on the geoscientific literature. The Journal of Geographical Sciences is a high-level academic journal in China (https://www.geog.com.cn/EN/home (accessed on 13 January 2024)) that publishes about 200 papers every year. We randomly selected 100 papers published in this journal from 2021 to 2023. From these articles, we extracted the paragraphs introducing the research data. Then, for each of the 100 pieces of text, we invited three experts to find and write down all of the geoscientific dataset names contained within it. A comparison showed that the three experts' results were almost identical. This result is used as the baseline to evaluate the ability of AI models to extract geoscientific dataset names.

The detailed process of extracting geoscientific dataset names is as follows: (1) For each of the 100 pieces of text, the NLPIR segmentation tool is used to split them into word sequences. The starting word of a piece of text is encoded by the method in Section 3.2.2 and the encoder is input to the trained HTM model. After processing by the classifier in the HTM model, the predicted encoder for the next word is output. Subsequently, the encoder for the actual next word is generated using the method described in Section 3.2.2. The similarity between the two encoders is calculated using the cosine similarity formula. If the similarity is greater than 0.4, the combination of the two adjacent words is deemed to conform to the naming pattern of geoscientific datasets, and they are part of a geoscientific dataset name. All words in the piece of text are dealt in the same way. If more than three consecutive words conformed to the pattern of scientific data naming, then the combination of these words is regarded as the name of a geoscientific dataset. (2) For GPT-4, we firstly use it directly to extract the geoscientific dataset names for every 100 pieces of text. The prompt for the zero-shot learning (ZSL) task is as follows: Please extract the names of the geoscientific datasets contained in the following paragraphs: {paragraphs containing geoscientific dataset names}. We then teach GPT-4 all 12,642 dataset names and use it again to extract the names of geoscientific datasets for every 100 pieces of text. In this context, for GPT-4, the prompt for the few-shot learning (FSL) task is as follows: The train-GeoscientificDatasetNames.txt file includes 12,642 geoscientific dataset names. Please learn the characteristics of these geoscientific dataset names. Now, based on the characteristics of the geoscientific dataset names mentioned and your knowledge, please extract the names of the geoscientific datasets contained in the following paragraphs: {paragraphs containing geoscientific dataset names}. The process of extracting geoscientific dataset names using GPT-4 is achieved by using the GPT-4 dialog window on the official website of OpenAI (https://openai.com/ (accessed on 25 March 2024)). (3) For Claude-3 (https://claude.ai (accessed on 5 June 2024)), we conduct similar experiments, which are tested on GPT-4.

To display our extraction results in an intuitive and representative way, we show the extraction results of two random pieces of text in Tables A2 and A3 in Appendix A. By comparing with the benchmark dataset names, it can be observed that our method can relatively accurately locate the positions of geoscientific dataset names and has relatively little redundancy when extracting specific geoscientific dataset names. However, when utilizing GPT-4 and Claude-3 for extracting geoscientific dataset names, issues such as redundancy or the omission of parts of words can occur. For instance, there are five correct geoscientific dataset names in Table A2, and our method can basically extract these names accurately. In contrast, the zero-shot learning (ZSL) method based on GPT-4 and Claude-3 can lose parts of the correct words of geoscientific dataset names. Both ZSL methods cannot correctly recognize "The spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m", and the GPT-4-based ZSL method cannot correctly recognize "1:1,000,000 soil type map". Additionally, the GPT-4-based ZSL and FSL methods can produce redundancy in recognizing geoscientific dataset names. For example, the correct geoscientific dataset name in Table A2 is "The Land Use and Cover Change (LUCC) dataset", but the recognized result is "The Land Use and Cover Change (LUCC) dataset from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences". The correct geoscientific dataset name in Table A3 is "The Fourth Forest Resources Inventory of Guangdong Province", but the recognized result is "The Fourth Forest Resources Inventory of Guangdong Province (partial pilot cities from 2013 to 2016, province-wide from 2017 to 2018)". Moreover, as shown in Tables A2 and A3, the ZSL method based on Claude-3 can extract more geoscientific datasets compared to the method based on GPT-4. However, it also demonstrates problems with redundancy and omission. GPT-4- and Claude-3-based methods extract more redundant or omissive results because the two methods may not understand the naming rules and combination patterns of various elements of geoscientific data. For example, the two methods will

extract geoscientific names containing the word "data" or "dataset," such as "NDVI data" and "precipitation dataset", which are not regarded as geoscientific dataset names because they do not include temporal or spatial coverage elements.

With all three methods, the extraction result contains three kinds of items: exact geoscientific dataset names, partially correct items (i.e., geoscientific dataset names with redundancy), and completely incorrect items. To quantitatively describe the extraction accuracy, we computed the precision, recall, and F1-score of the three methods.

Precision (P) represents the rate of correct results for all extracted dataset names, where T is the number of correctly extracted names and N is the number of all extracted names. The formula is as follows:

$$P = \frac{T}{N} \tag{12}$$

Recall (R) represents the ratio of correctly extracted geoscientific dataset names to the number of names in the benchmark, where T is the number of correctly extracted names and S is the number of names in the benchmark. The formula is as follows:

$$R = \frac{T}{S} \tag{13}$$

The F1-score is a composite indicator determined by a weighted average of precision and recall, which responds to the performance of the model. The formula is as follows:

$$F1\text{-score} = \frac{2PR}{(P+R)}. \tag{14}$$

The precision, recall, and F1-score of the three methods are shown in Table 4 and Figure 7. Comparison of the five methods on precision, recall, and F1-score.

Table 4. Statistics for the extraction results of the three methods for 100 randomly selected pieces of text.

Statistical Indicator	Proposed Method	GPT-4 Based Zero-Shot Learning (ZSL) Method	GPT-4 Based Few-Shot Learning (FSL) Method	Claude-3 Based Zero-Shot Learning (ZSL) Method	Claude-3 Based Few-Shot Learning (FSL) Method
Number of benchmark geoscientific dataset names	530	530	530	530	530
Number of extracted geoscientific dataset names	600	478	459	534	520
Number of correctly extracted geoscientific dataset names	411	340	345	368	378
Precision (%)	68.5	71.1	75.2	69.4	71.3
Recall (%)	77.5	64.2	65.1	68.9	72.7
F1-score	0.727	0.675	0.698	0.691	0.720

From Table 4 and Figure 7, it can be seen that GPT-4 and Claude-3 have better precision than our method because they extract fewer results. However, our method extracts the most correct geoscientific dataset names and performs better in terms of recall and F1-score. The precision of GPT-4 and Claude-3 is improved after few-shot learning (FSL). Our method can find more geoscientific dataset names, although some of them are not always exact, such as "data of precipitation, temperature and other daily observation data from national meteorological stations on the Qinghai Tibet Plateau", which is a partially correct name (the right name is "data of precipitation, temperature and other daily observation data from national meteorological stations on the Qinghai Tibet Plateau from 2020 to 2020") with a

rare combination of elements of geoscientific data. Overall, if the intention is to extract more correct datasets, researchers can consider using our method.

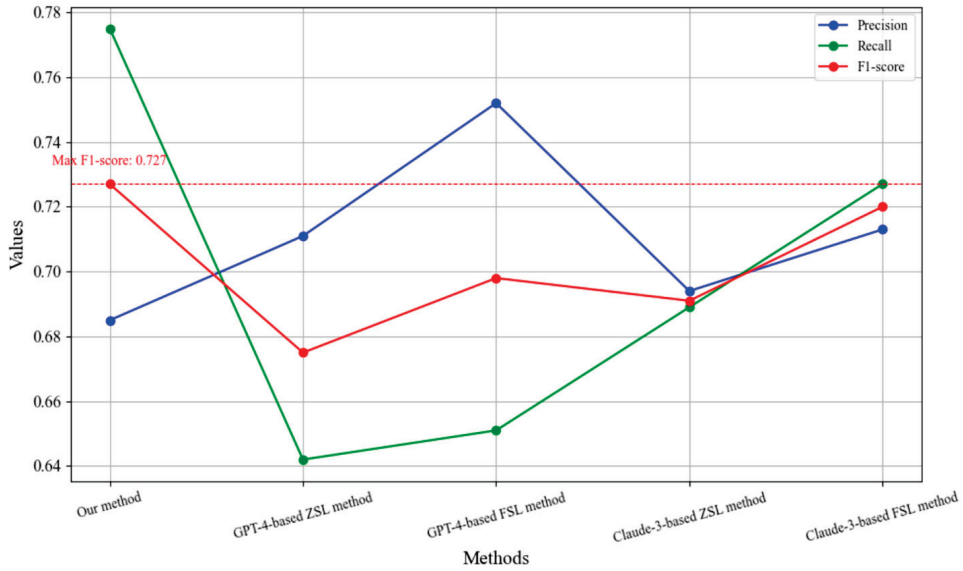


Figure 7. Comparison of the five methods on precision, recall, and F1-score.

6. Conclusions and Discussion

In this study, we propose an artificial neural network method for extracting geoscientific dataset names. Specifically, a new word-encoding method for the HTM model is proposed. The new word-encoding method uses Unicode to generate a binary vector for characters. The encoder is input into a human brain-inspired neural network (the HTM model) to generate a predictive vector. In our research, we ingeniously transform the predictive vector into the encoder of the next word by using the classifier of backpropagation (BP). By computing the similarity between the encoders of the next predictive word and the next real word, we can determine whether the combination of two adjacent words is part of a geoscientific dataset name and thus extract geoscientific dataset names from unstructured text. Through training with more than 12,000 dataset names, the HTM model learns various naming patterns and can be used to extract geoscientific dataset names from the literature. Finally, we compare our method with two existing methods—GPT-4 and Claude-3. The result shows that our method outperforms the existing methods in recall and F1-score.

Our new artificial neural network method for extracting geoscientific dataset names achieves a higher F1-score on the task of extracting the names of geoscientific datasets, while the large language model methods achieve higher accuracy because they extract fewer results. Our model requires a smaller training corpus and computing power than the large language models. Our method can be utilized not only for extracting knowledge from the literature and reproducing geoscientific studies but also for carrying out scientific data bibliometrics and other tasks related to geospatial information.

However, the method has the following limitations: First, while our encoding method demonstrates higher accuracy in encoding Chinese characters compared to the semantic folding method, the accuracy in encoding English characters needs improvement. Additionally, although the numerical mapping method reduces the encoding length, the length of encoded words remains relatively large, resulting in a longer training time for the HTM model. Second, while our method achieves a higher F1-score in extracting geoscientific dataset names compared to larger language models such as GPT-4, it has lower extraction accuracy. Moreover, as large language models learn more knowledge, their extraction accuracy continues to improve. Therefore, it is necessary to improve our method. In the future, we plan to modify existing word-embedding

techniques such as Word2vec to create a new word-encoding method for the HTM model. This method will be able to not only encode multiple languages but also reduce the encoding length and improve accuracy, thereby further improving the precision of the HTM model in extracting dataset names.

Author Contributions: Conceptualization, Zugang Chen, Xinqian Wu, and Kai Wu; methodology, Zugang Chen, Xinqian Wu, and Kai Wu; software, Zugang Chen, Kai Wu, and Hang Feng; validation, Kai Wu; formal analysis, Zugang Chen, Xinqian Wu, and Kai Wu; resources, Guoqing Li, Zugang Chen, and Xinqian Wu; data curation, Kai Wu and Haodong Wang; writing—original draft preparation, Kai Wu; writing—review and editing, Zugang Chen, Xinqian Wu, and Kai Wu; visualization, Kai Wu; supervision, Zugang Chen and Xinqian Wu; project administration, Zugang Chen and Xinqian Wu; funding acquisition, Zugang Chen, Jing Li, and Shaohua Wang. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 42201505), the Natural Science Foundation of Hainan Province of China (Grant No. 622QN352), and the National Key Research and Development Program of China (Grant No. 2021YFF070420304).

Data Availability Statement: The datasets and code can be found at https://github.com/WkStatisticsRoad/HTMTextExtract (accessed on 8 June 2024).

Acknowledgments: We thank the anonymous reviewers and all of the editors who participated in the revision process.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1 lists the main parameter settings for the hierarchical temporal memory (HTM) model utilized in this study. The parameters are input dimensions, number of columns, number of cells per column, potential radius, number of active columns, connected threshold for synaptic permanence, initial synaptic permanence, dendritic segment activation threshold, and synaptic permanence increment and decrement. In the experiment of extracting geoscientific dataset names, these parameters were employed to configure the HTM model. The specific explanation and setting of each parameter are as follows:

Table A1. Parameters of created HTM model.

Parameter Name	Value	Parameter Name	Value
Number of columns	5000	Number of cells per column	6
Input dimensions	22,476	Potential radius	6
Number of active columns	48	Connected threshold for synaptic permanence	0.7
Initial synaptic permanence	0.1	Dendritic segment activation threshold	4
Synaptic permanence increment	0.1	Synaptic permanence decrement	0.1

Input dimensions: This parameter represents the dimensionality of the input data, i.e., the number of input bits processed by the HTM model. We encoded 3596 commonly used Chinese characters and set the number of continuously active bits to 151, resulting in a character encoding length of 3746. In this experiment, the maximum character length of a word was set to six, resulting in a word-encoding length of 22,476, which was the input dimension for the HTM model.

Number of columns: This parameter determines the number of columns in the spatial pooler of the HTM model. Having more columns allows the HTM model to handle more features or patterns simultaneously, but also increases computational complexity. In this experiment, after tuning the parameters, the number of columns was set to 5000.

Number of cells per column: Each column in the spatial pooler contains a set of cells. This parameter specifies how many cells are contained in each column. Typically, the number of cells per column is not very large, usually ranging from a few to several dozen, as more cells will increase computational cost. In this experiment, based on experience and parameter optimization, the number of cells per column was set to six.

Potential radius: The potential radius defines the neighborhood size within which a cell in the spatial pooler can form connections. A larger potential radius allows cells to form connections with a broader area, but an excessively large potential radius can lead to overfitting. In this experiment, based on experience and parameter optimization, the potential radius was set to six.

Number of active columns: The number of active columns refers to the maximum number of columns that can remain active within a local inhibition area. By controlling the number of active columns, the sparsity and representation capacity of the HTM model can be adjusted. In this experiment, based on experience and parameter optimization, the number of active columns was set to 48.

Connected threshold for synaptic permanence: If the connected threshold for synaptic permanence (values in [0, 1]) is set to 0.7, initially 70% of the potential synapses are connected. If the permanence value for a synapse is greater than this threshold, it is considered connected. In this experiment, based on experience and parameter optimization, the connected threshold was set to 0.7.

Initial synaptic permanence: This parameter indicates the initial permanence value for a newly formed synapse. It is usually set to a low value to ensure that the initial connections are neither too strong nor too weak. In this experiment, the initial synaptic permanence was set to 0.1.

Dendritic segment activation threshold: This parameter indicates how many cells need to be active within a time step in order to activate the entire column. Its value is related to the number of cells in the column, and the threshold cannot exceed this number. In this experiment, based on experience and parameter optimization, the dendritic segment activation threshold was set to four.

Synaptic permanence increment: This parameter indicates how much a synapse is strengthened when the cell is activated. It is typically set to a positive value of less than 1 to ensure that synaptic connections gradually strengthen during learning. In this experiment, the synaptic permanence increment was set to 0.1.

Synaptic permanence decrement: This parameter indicates how much a synapse is weakened when the cell is inactivated. It is typically set to a positive value of less than one to ensure that inactive synaptic connections gradually weaken during learning. In this experiment, the synaptic permanence decrement was set to 0.1.

In summary, Table A1 describes the key parameters of the HTM model, including input dimensions, number of cells per column, and synaptic connectivity thresholds. These parameters are crucial in controlling the model's ability to encode input patterns, perform pattern recognition, and adapt to changes in the input data. Understanding and appropriately setting these parameters are essential for the effective application of the HTM model.

Table A2. Results of five methods for extracting geoscientific dataset names.

Extraction Method	Names of Benchmark Geoscientific Datasets	Results of Five Recognition Methods
Proposed method	The Land Use and Cover Change (LUCC) dataset, The China Annual Land Cover Dataset (CLCD), The regional data of China from GlobeLand30, The spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m, 1:1,000,000 soil type map	The data sources for this article are as follows: (1) Ecosystem-type data. **The Land Use and Cover Change (LUCC) dataset** from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences; **the China Annual Land Cover Dataset (CLCD)** based on satellite data from Google Earth Engine (GEE); and the **regional data of** China **from GlobeLand30**. Based on the aforementioned land use and cover types, they were transformed into ecosystem types such as cropland, forest, grassland, wetland, and settlement, obtaining the **spatial distribution data of the Qinghai-Tibet Plateau ecosystems** for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data. The soil property tables and spatial distribution data attached to the **1:1,000,000 soil type map** come from the Western Environmental and Ecological Science Data Center, and the soil erodibility factor was estimated using the Nomograph method.
GPT-4-based zero-shot learning (ZSL) method		The data sources for this article are as follows: (1) Ecosystem-type data. **The Land Use and Cover Change (LUCC) dataset** from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences; **the China Annual Land Cover Dataset (CLCD)** based on satellite data from Google Earth Engine (GEE); and **the regional data of China from GlobeLand30**. Based on the aforementioned land use and cover types, they were transformed into ecosystem types such as cropland, forest, grassland, wetland, and settlement, obtaining the spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data. The soil property tables and spatial distribution data attached to the 1:1,000,000 soil type map come from the Western Environmental and Ecological Science Data Center, and the soil erodibility factor was estimated using the Nomograph method.
GPT-4-based few-shot learning (FSL) method		The data sources for this article are as follows: (1) Ecosystem-type data. **The Land Use and Cover Change (LUCC) dataset from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences; the China Annual Land Cover Dataset (CLCD) based on satellite data from Google Earth Engine (GEE); and the regional data of China from GlobeLand30**. Based on the aforementioned land use and cover types, they were transformed into ecosystem types such as cropland, forest, grassland, wetland, and settlement, obtaining the spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data. The soil property tables and spatial distribution data attached to the **1:1,000,000 soil type map come from the Western Environmental and Ecological Science Data Center**, and the soil erodibility factor was estimated using the Nomograph method.
Proposed method	The Land Use and Cover Change (LUCC) dataset, The China Annual Land Cover Dataset (CLCD), The regional data of China from GlobeLand30, The spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m, 1:1,000,000 soil type map	Data sources for this article: (1) Ecosystem-type data: **The Land Use and Cover Change (LUCC) dataset** from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences; **the China Annual Land Cover Dataset (CLCD)** based on satellite data from Google Earth Engine (GEE); and the **regional data of** China **from GlobeLand30**. Based on land use and cover types, they were transformed into ecosystem types including cropland, forest, grassland, wetland, and settlement, to obtain **spatial distribution data of the Qinghai-Tibet Plateau ecosystems** for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data: Soil property tables and spatial distribution data attached to **1:1,000,000 soil type map** come from Western Environmental and Ecological Science Data Center, and soil erodibility factor was estimated using the nomograph method.
Claude-3-based zero-shot learning (ZSL) method		Data sources for this article: (1) Ecosystem-type data: **The Land Use and Cover Change (LUCC) dataset from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences**; the **China Annual Land Cover Dataset (CLCD) based on satellite data from Google Earth Engine (GEE)**; and **the regional data of China from GlobeLand30**. Based on land use and cover types, they were transformed into ecosystem types including cropland, forest, grassland, wetland, and settlement to obtain spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data: The soil property tables and spatial distribution data attached to **1:1,000,000 soil type map come from the Western Environmental and Ecological Science Data Center**, and the soil erodibility factor was estimated using the nomograph method.
Claude-3-based few-shot learning (FSL) method		Data sources for this article: (1) Ecosystem-type data: **The Land Use and Cover Change (LUCC) dataset from the Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences**; the **China Annual Land Cover Dataset (CLCD) based on satellite data from Google Earth Engine (GEE)**; and **the regional data of China from GlobeLand30**. Based on land use and cover types, they were transformed into ecosystem types including cropland, forest, grassland, wetland, and settlement, to obtain spatial distribution data of the Qinghai-Tibet Plateau ecosystems for the years 2000, 2010, and 2020 with a spatial resolution of 30 m. (2) Soil data: **The soil property tables and spatial distribution data** attached to the **1:1,000,000 soil type map** come **from the Western Environmental and Ecological Science Data Center**, and soil erodibility factor was estimated using the nomograph method.

Table A3. Results of five methods for extracting geoscientific dataset names.

Extraction Method	Names of Benchmark Geoscientific Datasets	Results of Five Recognition Methods
Proposed method	The Fourth Forest Resources Inventory of Guangdong Province, The second National Soil Survey Data for Guangdong (1979–1985), Harmonized World Soil Database, The basic attribute dataset of China's high-resolution national soil information grid for Guangdong (2010–2018)	**The Fourth Forest Resources Inventory of Guangdong Province** (partial pilot cities 2013–2016, province-wide 2017–2018) is sourced from Guangdong Provincial Department of Natural Resources in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 km^2 each. **The second National Soil Survey data for Guangdong** (1979–1985), with information such as soil organic carbon content and soil bulk density, is sourced from the **Harmonized World Soil Database** by the UN Food and Agriculture Organization, with a spatial resolution of 0.25 km × 0.25 km. **The basic attribute dataset of** China's **high-resolution national soil information grid for Guangdong (2010–2018)** also provides information on soil organic carbon content and bulk density, sourced from the Soil Data Center of the National Earth System Science Data Center (http://soil.geodata.cn/ (accessed on 12 April 2022)), with a spatial resolution of 1 km × 1 km.
GPT-4-based zero-shot learning (ZSL) method		**The Fourth Forest Resources Inventory of Guangdong Province (partial pilot cities 2013–2016, province-wide 2017–2018)** was sourced from the Guangdong Provincial Department of Natural Resources in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 km^2 each. **The second National Soil Survey data for Guangdong (1979–1985)**, with information including soil organic carbon content and soil bulk density, sourced from the Harmonized World Soil Database by the UN Food and Agriculture Organization, with spatial resolution of 0.25 km × 0.25 km. **The basic attribute dataset of China's high-resolution national soil information grid for Guangdong (2010–2018)** also provides information on soil organic carbon content and bulk density, sourced from the Soil Data Center of the National Earth System Science Data Center (http://soil.geodata.cn/ (accessed on 12 April 2022)), with a spatial resolution of 1 km × 1 km.
GPT-4-based few-shot learning (FSL) method		**The Fourth Forest Resources Inventory of Guangdong Province** (partial pilot cities 2013–2016, province-wide 2017–2018) sourced from the Guangdong Provincial Department of Natural Resources in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 km^2 each. **The second National Soil Survey data for Guangdong** (1979–1985), with information including soil organic carbon content and bulk density, sourced from **the Harmonized World Soil Database by the UN Food and Agriculture Organization**, with spatial resolution of 0.25 km × 0.25 km. **The basic attribute dataset of China's high-resolution national soil information grid for Guangdong** (2010–2018) also provides information on soil organic carbon content and bulk density, sourced from the **Soil Data Center of the National Earth System Science Data Center** (http://soil.geodata.cn/ (accessed on 12 April 2022)), with spatial resolution of 1 km × 1 km.
Proposed method	The Fourth Forest Resources Inventory of Guangdong Province, The second National Soil Survey Data for Guangdong (1979–1985), Harmonized World Soil Database, The basic attribute dataset of China's high-resolution national soil information grid for Guangdong (2010–2018)	**The Fourth Forest Resources Inventory of Guangdong Province** (partial pilot cities 2013–2016, province-wide 2017–2018) sourced from Guangdong Provincial Department of Natural Resources in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 km^2 each. **The second National Soil Survey data for Guangdong** (1979–1985), with information including soil organic carbon content and bulk density, sourced from the **Harmonized World Soil Database** by the UN Food and Agriculture Organization, with spatial resolution of 0.25 km × 0.25 km. **The basic attribute dataset of** China's **high-resolution national soil information grid for Guangdong (2010–2018)** also provides information on soil organic carbon content and bulk density, sourced from the Soil Data Center of the National Earth System Science Data Center (http://soil.geodata.cn/ (accessed on 12 April 2022)), with spatial resolution of 1 km × 1 km.
Claude-3-based zero-shot learning (ZSL) method		The **Fourth Forest Resources Inventory of Guangdong Province (partial pilot cities 2013–2016, province-wide 2017–2018) sourced from Guangdong Provincial Department of Natural Resources** in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 km^2 each. The **second National Soil Survey data for Guangdong (1979–1985)**, with information including soil organic carbon content and bulk density, sourced from **the Harmonized World Soil Database by the UN Food and Agriculture Organization**, with spatial resolution of 0.25 km × 0.25 km. The basic attribute dataset of **China's high-resolution national soil information grid for Guangdong (2010–2018)** also provides information on soil organic carbon content and bulk density, sourced **from the Soil Data Center of the National Earth System Science Data Center** (http://soil.geodata.cn/ (accessed on 12 April 2022)), with spatial resolution of 1 km × 1 km.
Claude-3-based few-shot learning (FSL) method		**The Fourth Forest Resources Inventory of Guangdong Province** (partial pilot cities 2013–2016, province-wide 2017–2018) sourced from Guangdong Provincial Department of Natural Resources in the form of vector layers, with precision to the forestry sub compartment scale. Guangdong Province has 2,403,557 forestry sub compartments, with an average area of about 0.07 km^2 each. **The second National Soil Survey data for Guangdong (1979–1985)**, with information including soil organic carbon content and bulk density, **sourced from the Harmonized World Soil Database by the UN Food and Agriculture Organization**, with spatial resolution of 0.25 km × 0.25 km. **The basic attribute dataset of China's high-resolution national soil information grid for Guangdong (2010–2018)** also provides information on soil organic carbon content and bulk density, **sourced from the Soil Data Center of the National Earth System Science Data Center** (http://soil.geodata.cn/ (accessed on 12 April 2022)), with spatial resolution of 1 km × 1 km.

To display our extraction results in a more intuitive and representative way, we randomly selected the extraction results from two paragraphs. Tables A2 and A3 in Appendix A show part of the recognition results of the three methods (Represent the recognized results in bold black text). From these tables, it can be observed that the proposed method can relatively accurately locate the positions of geoscientific dataset names and has relatively low redundancy when identifying specific geoscientific dataset names. A detailed analysis can be found in Section 5.2.

References

1. Li, J.; Zhou, C. Analysis on the Characteristics of Geospatial Data. *Sci. Geogr. Sin.* **1999**, *19*, 158–162. [CrossRef]
2. Lu, M.; Appel, M.; Pebesma, E. Multidimensional Arrays for Analysing Geoscientific Data. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 313. [CrossRef]
3. Buttlar, J.v.; Zscheischler, J.; Mahecha, M.D. An extended approach for spatiotemporal gapfilling: Dealing with large and systematic gaps in geoscientific datasets. *Nonlin. Process. Geophys.* **2014**, *21*, 203–215. [CrossRef]
4. Sun, K.; Zhu, Y.; Pan, P.; Hou, Z.; Wang, D.; Li, W.; Song, J. Geospatial data ontology: The semantic foundation of geospatial data integration and sharing. *Big Earth Data* **2019**, *3*, 269–296. [CrossRef]
5. Kostoff, R.N. Role of Technical Literature in Science and Technology Development and Exploitation. *J. Inf. Sci.* **2003**, *29*, 223–228. [CrossRef]
6. Ning, B.; Zhao, Y. To Embrace Open Science More Closely. *Innovation* **2020**, *1*, 100012. [CrossRef] [PubMed]
7. Morse, P.; Reading, A.; Lueg, C. Animated analysis of geoscientific datasets: An interactive graphical application. *Comput. Geosci.* **2017**, *109*, 87–94. [CrossRef]
8. Konkol, M.; Kray, C.; Pfeiffer, M. Computational reproducibility in geoscientific papers: Insights from a series of studies with geoscientists and a reproduction study. *Int. J. Geogr. Inf. Sci.* **2019**, *33*, 408–429. [CrossRef]
9. Gil, Y.; David, C.H.; Demir, I.; Essawy, B.T.; Fulweiler, R.W.; Goodall, J.L.; Karlstrom, L.; Lee, H.; Mills, H.J.; Oh, J.H.; et al. Toward the Geoscience Paper of the Future: Best practices for document ing and sharing research from data to software to provenance. *Earth Space Sci.* **2016**, *3*, 388–415. [CrossRef]
10. Zhang, S.; Xu, H.; Jia, Y.; Wen, Y.; Wang, D.; Fu, L.; Wang, X.; Zhou, C. GeoDeepShovel: A platform for building scientific database from geoscience literature with AI assistance. *Geosci. Data J.* **2023**, *10*, 519–537. [CrossRef]
11. Tao, L.; Xie, Z.; Xu, D.; Ma, K.; Qiu, Q.; Pan, S.; Huang, B. Geographic Named Entity Recognition by Employing Natural Language Processing and an Improved BERT Model. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 598. [CrossRef]
12. Chung, C.-J.F.; Fabbri, A.G. The representation of geoscience information for data integration. *Nonrenew. Resour.* **1993**, *2*, 122–139. [CrossRef]
13. Arias, A.; Dini, I.; Casini, M.; Fiordelisi, A.; Perticone, I.; Pisano, A. Geoscientific Feature Update of the Larderello-Travale Geothermal System (Italy) for a Regional Numerical Modeling. In Proceedings of the World Geothermal Congress 2010, Bali, Indonesia, 25–30 April 2010.
14. Färber, M.; Albers, A.; Schüber, F. Identifying Used Methods and Datasets in Scientific Publications. In Proceedings of the SDU@AAAI Workshop on Scientific Document Understanding, Online, 19 February 2021.
15. Heddes, J.; Meerdink, P.; Pieters, M.; Marx, M. The Automatic Detection of Dataset Names in Scientific Articles. *Data* **2021**, *6*, 84. [CrossRef]
16. George, D.; Hawkins, J. A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005.
17. George, D.; Hawkins, J. Towards a Mathematical Theory of Cortical Micro-circuits. *PLoS Comput. Biol.* **2009**, *5*, e1000532. [CrossRef] [PubMed]
18. Klukas, M.; Lewis, M.; Fiete, I. Efficient and flexible representation of higher-dimensional cognitive variables with grid cells. *PLOS Comput. Biol.* **2020**, *16*, e1007796. [CrossRef]
19. Cao, Q.; Wang, S.; Chen, Z.; Li, G.; Li, J. The Method of Extracting Names of Geo-science Data based on Regular Expressions. *J. Geo-Inf. Sci.* **2023**, *25*, 1601–1610. [CrossRef]
20. Afzal, M.T.; Maurer, H.A.; Balke, W.-T.; Kulathuramaiyer, N. Rule based Autonomous Citation Mining with TIERL. *J. Digit. Inf. Manag.* **2010**, *8*, 196–204.
21. Fries, J.A.; Varma, P.; Chen, V.S.; Xiao, K.; Tejeda, H.; Saha, P.; Dunnmon, J.A.; Chubb, H.; Maskatia, S.A.; Fiterau, M.; et al. Weakly supervised classification of aortic valve malformations using unlabeled cardiac MRI sequences. *Nat. Commun.* **2019**, *10*, 3111. [CrossRef] [PubMed]
22. Soni, A.; Viswanathan, D.; Pachaiyappan, N.; Natarajan, S. A Comparison of Weak Supervision methods for Knowledge Base Construction. In Proceedings of the AKBC@NAACL-HLT, San Diego, CA, USA, 12–17 June 2016.
23. Zech, J.R.; Pain, M.; Titano, J.J.; Badgeley, M.A.; Schefflein, J.; Su, A.; Costa, A.B.; Bederson, J.B.; Lehár, J.; Oermann, E.K. Natural Language-based Machine Learning Models for the Annotation of Clinical Radiology Reports. *Radiology* **2018**, *287*, 570–580. [CrossRef]

24. Cui, B.-G.; Chen, X. An Improved Hidden Markov Model for Literature Metadata Extraction. In Proceedings of the 6th International Conference on Advanced Intelligent Computing Theories and Applications: Intelligent Computing, Changsha, China, 18 August 2010; pp. 205–212.
25. Zhang, K.; Xu, H.; Tang, J.; Li, J.-Z. Keyword Extraction Using Support Vector Machine. In Proceedings of the Interational Conference on Web-Age Information Management, Hong Kong, China, 17–19 June 2006; pp. 85–96.
26. Kaur, J.; Gupta, V. Effective Approaches for Extraction of Keywords. *Int. J. Comput. Sci.* **2010**, *7*, 144–148.
27. Han, H.; Giles, C.L.; Manavoglu, E.; Zha, H.; Zhang, Z.; Fox, E.A. Automatic document metadata extraction using support vector machines. In Proceedings of the 2003 Joint Conference on Digital Libraries, Houston, TX, USA, 20 June 2003; pp. 37–48.
28. Shinde, P.; Shah, S. A Review of Machine Learning and Deep Learning Applications. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 25 April 2018; pp. 1–6.
29. Zhao, Z.; Yang, Z.; Luo, L.; Wang, L.; Zhang, Y.; Lin, H.; Wang, J. Disease named entity recognition from biomedical literature using a novel convolutional neural network. *BMC Med. Genom.* **2017**, *10*, 73. [CrossRef]
30. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv* **2015**, arXiv:1508.01991.
31. Delgado, J.; Ebreso, U.; Kumar, Y.; Li, J.J.; Morreale, P. Preliminary Results of Applying Transformers to Geoscience and Earth Science Data. In Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2022; pp. 284–288.
32. Bhattarai, K.; Oh, I.; Sierra, J.; Payne, P.; Abrams, Z.; Lai, A. Leveraging GPT-4 for Identifying Clinical Phenotypes in Electronic Health Records: A Performance Comparison between GPT-4, GPT-3.5-turbo and spaCy's Rule-based & Machine Learning-based methods. *bioRxiv* **2023**. *preprint*. [CrossRef]
33. Yao, R.; Hou, L.; Ye, Y.; Zhang, J.; Wu, J. Method and Dataset Mining in Scientific Papers. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 6260–6262.
34. Kumar, S.; Ghosal, T.; Ekbal, A. DataQuest: An Approach to Automatically Extract Dataset Mentions from Scientific Papers. In Proceedings of the International Conference on Asian Digital Libraries, Hanoi, Vietnam, 30 November 2021; pp. 43–53.
35. Younes, Y.; Scherp, A. Question Answering Versus Named Entity Recognition for Extracting Unknown Datasets. *IEEE Access* **2023**, *11*, 92775–92787. [CrossRef]
36. Beltagy, I.; Lo, K.; Cohan, A. SciBERT: A Pretrained Language Model for Scientific Text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3606–3611.
37. Lin, Z.; Deng, C.; Zhou, L.; Zhang, T.; Xu, Y.; Xu, Y.; He, Z.; Shi, Y.; Dai, B.; Song, Y.; et al. GeoGalactica: A Scientific Large Language Model in Geoscience. *arXiv* **2023**, arXiv:2401.00434.
38. Ahsan, M.M.T.; Rahaman, M.S.; Anjum, N. From ChatGPT-3 to GPT-4: A Significant Leap in AI-Driven NLP Tools. *J. Eng. Emerg. Technol.* **2023**, *1*, 50–60. [CrossRef]
39. Hosseini, S.; Najafipour, S.; Cheung, N.-M.; Yin, H.; Kangavari, M.R.; Zhou, X. TEAGS: Time-aware text embedding approach to generate subgraphs. *Data Min. Knowl. Discov.* **2020**, *34*, 1136–1174. [CrossRef]
40. Najafipour, S.; Hosseini, S.; Hua, W.; Kangavari, M.R.; Zhou, X. SoulMate: Short-Text Author Linking Through Multi-Aspect Temporal-Textual Embedding. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 448–461. [CrossRef]
41. Hosseini, S.; Yin, H.; Zhou, X.; Sadiq, S.; Kangavari, M.R.; Cheung, N.-M. Leveraging multi-aspect time-related influence in location recommendation. *World Wide Web* **2019**, *22*, 1001–1028. [CrossRef]
42. Hosseini, S.; Yin, H.; Zhang, M.; Zhou, X.; Sadiq, S. Jointly Modeling Heterogeneous Temporal Properties in Location Recommendation. In *Database Systems for Advanced Applications*; Springer: Cham, Switzerland, 2017; pp. 490–506.
43. Saaki, M.; Hosseini, S.; Rahmani, S.; Kangavari, M.R.; Hua, W.; Zhou, X. Value-Wise ConvNet for Transformer Models: An Infinite Time-Aware Recommender System. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 9932–9945. [CrossRef]
44. Malawade, A.; Costa, N.; Muthirayan, D.; Khargonekar, P.; Al Faruque, M.A. Neuroscience-Inspired Algorithms for the Predictive Maintenance of Manufacturing Systems. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7980–7990. [CrossRef]
45. Zeng, H.; Zhao, X.; Wang, L. Multivariate Time Series Anomaly Detection On Improved HTM Model. In Proceedings of the 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), Fuzhou, China, 24–26 September 2021; pp. 759–763.
46. Krestinskaya, O.; Ibrayev, T.; James, A.P. Hierarchical Temporal Memory Features with Memristor Logic Circuits for Pattern Recognition. *IEEE Trans. Comput. -Aided Des. Integr. Circuits Syst.* **2018**, *37*, 1143–1156. [CrossRef]
47. Irmanova, A.; Krestinskaya, O.; James, A.P. Image Based HTM Word Recognizer for Language Processing. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics—Asia (ICCE-Asia), Jeju, Republic of Korea, 24–26 June 2018; pp. 206–212.
48. Almehmadi, A.; Bosakowski, T.; Sedky, M.; Bastaki, B.B. HTM Based Anomaly Detecting Model for Traffic Congestion. In Proceedings of the 2020 4th International Conference on Cloud and Big Data Computing, Virtual, UK, 26–28 August 2020; pp. 97–101.
49. Szoplák, Z.; Andrejková, G. Anomaly Detection in Text Documents using HTM Networks. In Proceedings of the Conference on Theory and Practice of Information Technologies, Muran, Slovakia, 24–28 September 2021; pp. 20–28.

50. Khan, H.M.; Khan, F.M.; Khan, A.; Asghar, M.Z.; Alghazzawi, D.M. Anomalous Behavior Detection Framework Using HTM-Based Semantic Folding Technique. *Comput. Math. Methods Med.* **2021**, *2021*, 5585238. [CrossRef] [PubMed]
51. Mackenzie, J.; Roddick, J.F.; Zito, R. An Evaluation of HTM and LSTM for Short-Term Arterial Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1847–1857. [CrossRef]
52. Zyarah, A.M.; Kudithipudi, D. Neuromorphic Architecture for the Hierarchical Temporal Memory. *IEEE Trans. Emerg. Top. Comput. Intell.* **2019**, *3*, 4–14. [CrossRef]
53. Hawkins, J.; George, D.; Niemasik, J. Sequence memory for prediction, inference and behaviour. *Philos. Trans. R. Soc. Lond. Ser. B Biol. Sci.* **2009**, *364*, 1203–1209. [CrossRef]
54. Kostavelis, I.; Gasteratos, A. On the optimization of Hierarchical Temporal Memory. *Pattern Recognition Letters* **2012**, *33*, 670–676. [CrossRef]
55. Zhou, L.; Zhang, D. NLPIR: A Theoretical Framework for Applying Natural Language Processing to Information Retrieval. *JASIST* **2003**, *54*, 115–123. [CrossRef]
56. Hawkins, J.; George, D. *Hierarchical Temporal Memory Concepts, Theory, and Terminology*; Numenta: Redwood City, CA, USA, 2006.
57. Hawkins, J.; Ahmad, S. Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex. *Front. Neural Circuits* **2016**, *10*, 23. [CrossRef]
58. Cui, Y.; Ahmad, S.; Hawkins, J. The HTM Spatial Pooler—A Neocortical Algorithm for Online Sparse Distributed Coding. *Front. Comput. Neurosci.* **2017**, *11*, 111. [CrossRef]
59. Kanerva, P. Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors. *Cogn. Comput.* **2009**, *1*, 139–159. [CrossRef]
60. Purdy, S. Encoding data for HTM systems. *arXiv* **2016**, arXiv:1602.05925. [CrossRef]
61. Bettels, J.; Bish, F.A. Unicode: A Universal Character Code. *Digit. Tech. J. Digit. Equip. Corp.* **1993**, *5*, 21–31.
62. Allen, J.D.; Anderson, D.; Becker, J.; Cook, R.; Davis, M.; Edberg, P.; Everson, M.; Freytag, A.; Iancu, L.; Ishida, R.; et al. *The Unicode Standard, Version 7.0*; Unicode: Mountain View, CA, USA, 2014.
63. Cui, Y.; Ahmad, S.; Hawkins, J. Continuous online sequence learning with an unsupervised neural network model. *Neural Comput.* **2016**, *28*, 2474–2504. [CrossRef] [PubMed]
64. Niu, D.; Yang, L.; Cai, T.; Li, L.; Wu, X.; Wang, Z. A New Hierarchical Temporal Memory Algorithm Based on Activation Intensity. *Comput. Intell. Neurosci.* **2022**, *2022*, 6072316. [PubMed]
65. Wielgosz, M.; Pietroń, M. Using Spatial Pooler of Hierarchical Temporal Memory to classify noisy videos with predefined complexity. *Neurocomputing* **2017**, *240*, 84–97. [CrossRef]
66. Wright, L.G.; Onodera, T.; Stein, M.M.; Wang, T.; Schachter, D.T.; Hu, Z.; McMahon, P.L. Deep physical neural networks trained with backpropagation. *Nature* **2021**, *601*, 549–555. [CrossRef]
67. Wen, J.; Zhao Jia, L.; Luo Si, W.; Han, Z. The improvements of BP neural network learning algorithm. In Proceedings of the WCC 2000—ICSP 2000 5th International Conference on Signal Processing Proceedings and 16th World Computer Congress 2000, Beijing, China, 21–25 August 2000; pp. 1647–1649.
68. Webber, F.D.S. Semantic Folding Theory And its Application in Semantic Fingerprinting. *arXiv* **2015**, arXiv:1511.08855. [CrossRef]
69. Chen, Z.; Song, J.; Yang, Y. An Approach to Measuring Semantic Relatedness of Geographic Terminologies Using a Thesaurus and Lexical Database Sources. *ISPRS Int. J. Geo Inf.* **2018**, *7*, 98.
70. Greenland, S.; Senn, S.J.; Rothman, K.J.; Carlin, J.B.; Poole, C.; Goodman, S.N.; Altman, D.G. Statistical tests, P values, confidence intervals, and power: A guide to misinterpretations. *Eur. J. Epidemiol.* **2016**, *31*, 337–350.
71. Wang, S.; Sun, X.; Li, X.; Ouyang, R.; Wu, F.; Zhang, T.; Li, J.; Wang, G. GPT-NER: Named Entity Recognition via Large Language Models. *arXiv* **2023**, arXiv:2304.10428. [CrossRef]
72. Ashok, D.; Lipton, Z.C. PromptNER: Prompting for Named Entity Recognition. *arXiv* **2023**, arXiv:2305.15444. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

An Improved ANN-Based Label Placement Method Considering Surrounding Features for Schematic Metro Maps

Zhiwei Wu, Tian Lan *, Chenzhen Sun, Donglin Cheng, Xing Shi, Meisheng Chen and Guangjun Zeng

Faculty of Geosciences and Engineering, Southwest Jiaotong University, Chengdu 611756, China; wuzhiwei@my.swjtu.edu.cn (Z.W.); immortal000@my.swjtu.edu.cn (C.S.); dlcheng@my.swjtu.edu.cn (D.C.); 2023201211@my.switu.edu.cn (X.S.); 2021114683@my.swjtu.edu.cn (M.C.); zenggj@my.swjtu.edu.cn (G.Z.)
* Correspondence: tianlan@swjtu.edu.cn

Abstract: On schematic metro maps, high-quality label placement is helpful to passengers performing route planning and orientation tasks. It has been reported that the artificial neural network (ANN) has the potential to place labels with learned labeling knowledge. However, the previous ANN-based method only considered the effects of station points and their connected edges. Indeed, unconnected but surrounding features (points, edges, and labels) also significantly affect the quality of label placement. To address this, we have proposed an improved method. The relations between label positions and both connected and surrounding features are first modeled based on labeling natural intelligence (i.e., the experience, knowledge, and rules of labeling established by cartographers). Then, ANN is employed to learn such relations. Quantitative evaluations show that our method reaches lower percentages of label–point overlap (0.00%), label–edge overlap (4.12%), and label–label overlap (20.58%) compared to the benchmark (4.17%, 14.29%, and 35.11%, respectively). On the other hand, our method effectively avoids ambiguous labels and ensures labels from the same line are placed on the same side. Qualitative evaluations show that approximately 75% of users prefer our results. This novel method has the potential to advance the automated generation of schematic metro maps.

Keywords: schematic metro maps; label placement; artificial neural network; natural intelligence

1. Introduction

Schematic metro maps are abstract representations of metro networks where the complexity is reduced by removing geographical features, simplifying lines, reorienting lines, etc. [1,2]. With such maps, passengers can quickly and accurately perform metro route planning and orientation tasks. One prominent example is the London Underground map (see Figure 1), where simplified metro lines are in vertical, horizontal, and diagonal directions. On the other hand, the congested areas have been expanded, and most details of geographic features have been removed. At present, these design principles usually serve as a basis for guiding automatic schematic metro map design [3].

Generally, the automatic schematic metro map design is divided into two subtasks: metro map layout and metro map labeling [4–8]. The research on metro map layout has received much attention and achieved remarkable results. In the early stage, a series of criteria for map layouts have been defined, and some methods for automatically generating schematic maps have been proposed with these criteria [4,9,10]. Later, researchers aimed to improve metro map layouts from different perspectives. It has been pointed out that the edge lengths should be maintained equal to ensure regular spacing between stations [1,11,12]. On the other hand, the directions of the edge are commonly along vertical, horizontal, and 45-degree diagonals, named octolinearity [13]. Recently, it has been found that other direction design styles (e.g., multilinear and curvilinear) may be preferred by passengers in some scenes [2,13]. Moreover, the topological relation between features and the main structure of the network should be preserved for high-quality schematic metro maps [14–18].

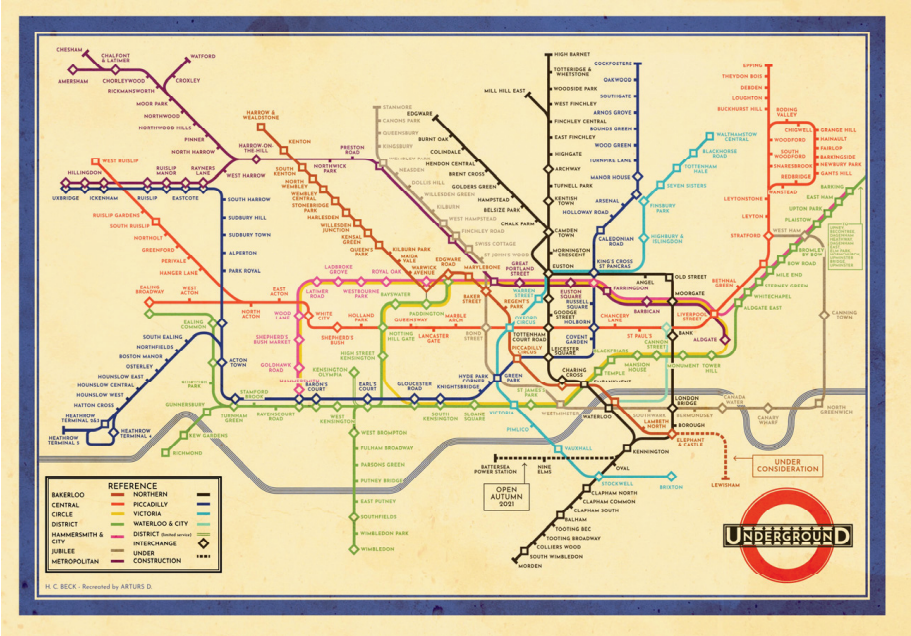
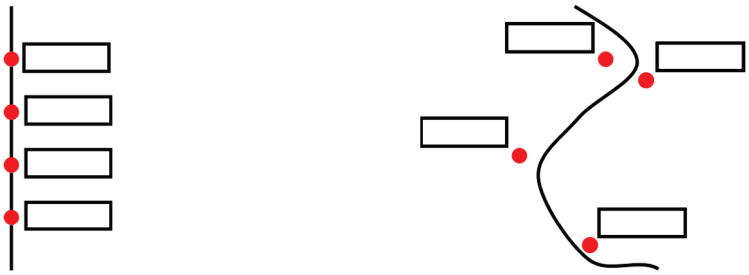


Figure 1. Schematic metro map of London was designed by Harry Beck in the 1930s [19].

The research on metro map labeling has received relatively limited attention [1,6,7,11]. In theory, the automatic labeling method should be developed based on the labeling natural intelligence of schematic metro maps (i.e., the experience, knowledge, and rules in labeling established by cartographers). In practice, such labeling natural intelligence is complex, and it is challenging to model them with mathematical expressions. The point labeling natural intelligence on topographic maps (which have formed a series of mathematical principles [20–22]) is commonly used as an alternative [11]. However, the applicability of these principles to schematic metro maps may be inappropriate in some situations [6]. For example, one crucial aspect of metro map labeling states: "labels of those points touching the same passing line are placed together on the same side" (see Figure 2a). By contrast, on topographic maps, points are usually positioned adjacent to rather than touching the line, and one labeling principle states: "a label and the corresponding point should be placed at the same side" (see Figure 2b).



(**a**) Labeling result of schematic metro maps (**b**) Labeling result of topographic maps

Figure 2. Comparison of labeling results between (**a**) schematic metro maps and (**b**) topographic maps.

The modeling of labeling natural intelligence for schematic metro maps has been reconsidered in recent years [6,7,23]. Labeling natural intelligence has been modeled from official schematic metro maps in two ways: to count labeling positions for point–edge situations in a statistical manner [6] and to learn the characteristics of point–edge situations by an artificial neural network (ANN) [7]. The first way depends on the pre-defined point–edge situations. In other words, if a specific relation is not pre-defined, it may be difficult to identify appropriate labeling positions for this relation. Theoretically, this problem can be solved by pre-defining all relations, but it requires a lot of time and resources. The

second way employs an artificial neural network (with good generalization ability) to learn the characteristics of point–edge situations and infer labeling positions with examples. Although it is a good idea, the basic unit used for learning, i.e., point–edge situations, only considers the point referred to by the label and the edges connected to the point. Indeed, the unconnected but surrounding points and edges also significantly affect the quality of label placement. Figure 3a shows the fixed label position of a point–edge situation, and no overlaps are observed. However, overlaps occur in Figure 3b when other features (points and edges in this case) surround this fixed label. To address this, we aim to develop an improved labeling method by integrating relations between label positions and both connected and surrounding features into the ANN model.

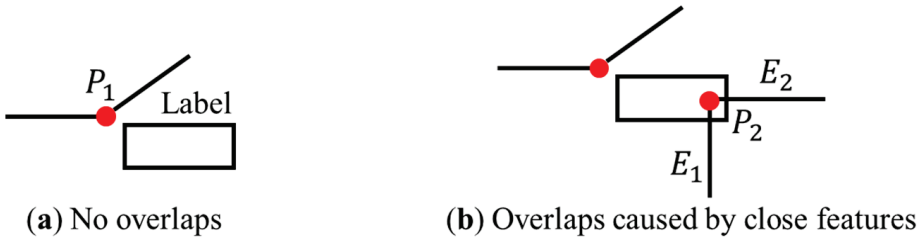


Figure 3. Limitation of label placement based on a point–edge situation.

The remainder of this article is organized as follows: In Section 2, we analyze and model the labeling natural intelligence of schematic metro maps. In Section 3, we propose an ANN-based label placement method with labeling natural intelligence. In Section 4, the experimental evaluation is presented. Some discussion and conclusions are made in Sections 5 and 6, respectively.

2. Analysis and Modeling of Labeling Natural Intelligence

To learn the new relations, we first conduct a thorough analysis of labeling natural intelligence. Then, we model them with a set of variables.

2.1. Analysis of Labeling Natural Intelligence

Through a thorough survey of the literature on metro map labeling (see Table 1), we have identified four common labeling natural intelligences: (1) place labels without overlaps; (2) place labels without ambiguity; (3) place labels from the same passing line on the same side; and (4) place labels in the most preferred positions.

Table 1. Common labeling natural intelligence in the existing methods.

Labeling Natural Intelligence	Source
Place labels without overlaps	References [1,4–7,9,11,12,24–30]
Place labels without ambiguity	References [1,6,11,12,24,29,30]
Place labels from the same passing line on the same side	References [1,5–7,11,12,24,26–28,30]
Place labels in the most preferred position	References [1,4,6,7,9,11,12,26–28,30]

2.1.1. Place Labels without Overlaps

The avoidance of overlapping labels is the most basic and important aspect [1,10], which aims to ensure the clarity and legibility of the map. Overlapping labels can be categorized into three kinds of cases: label–point overlap, label–edge overlap, and label–label overlap, as shown in Figure 4.

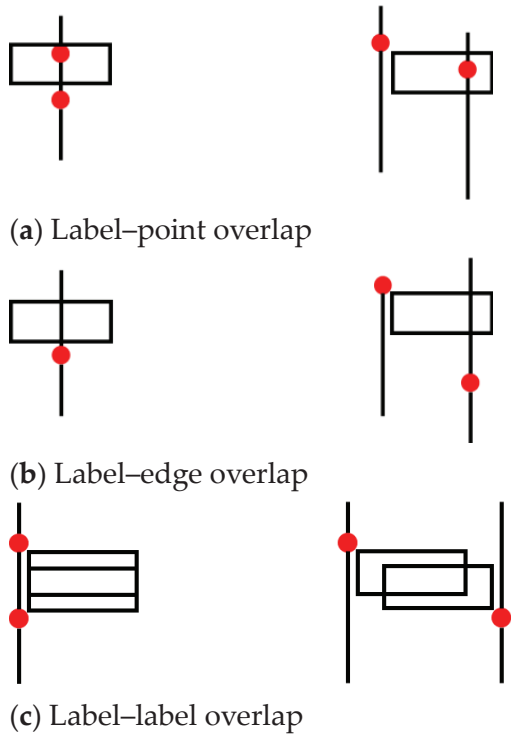


Figure 4. Three types of overlapping labels in schematic metro maps.

2.1.2. Place Labels without Ambiguity

Labels are often the only way to identify stations on a metro map. Therefore, ambiguous associations must be forbidden to ensure that passengers can accurately identify the location of each station. Two examples of ambiguous labels are shown in Figure 5. In these examples, it is unclear which point the labels associate.

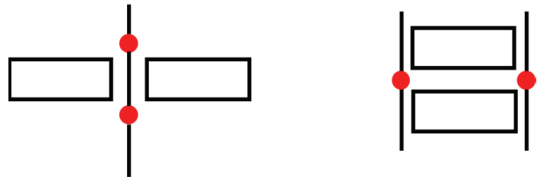


Figure 5. Examples of ambiguous labels.

2.1.3. Place Labels from the Same Passing Line on the Same Side

To maintain consistency and continuity on the metro maps, it is necessary to place labels for points touching the same passing line on the same side as much as possible, as shown in Figure 6. Additionally, if the labels are arbitrarily placed on the two sides of the passing line, this can lead to visual confusion and make the map difficult to read.

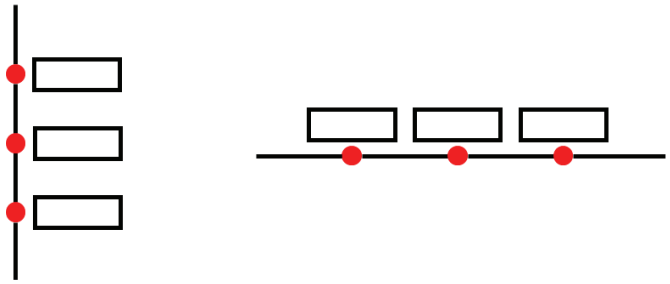


Figure 6. Label placements for points touching the same passing line are placed on the same side.

2.1.4. Place Labels in the Most Preferred Position

Labels should be placed in the most preferred position among all acceptable positions. Such a preferred position means the visual and cognitive preferences when viewing schematic metro maps. Some examples of preferred label placements are shown in Figure 7 (see reference [6] for the preferred label positions of all point–edge situations).

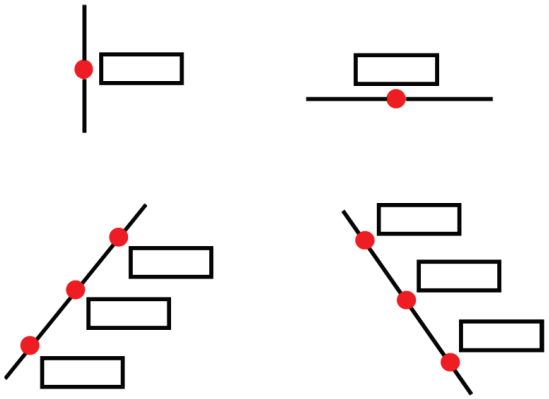


Figure 7. Examples of preferred label placements.

Conflicts may be caused among the four labeling natural intelligences mentioned above. In Figure 8a, all labels are placed on the right of the passing line, and this satisfies the third labeling natural intelligence but violates the first one. Indeed, such overlapped results are most unacceptable, and some possible improvements (e.g., Figure 8b,c) will be made by designers. This indicates that there exists a labeling natural intelligence that determines the prioritization among labeling principles.

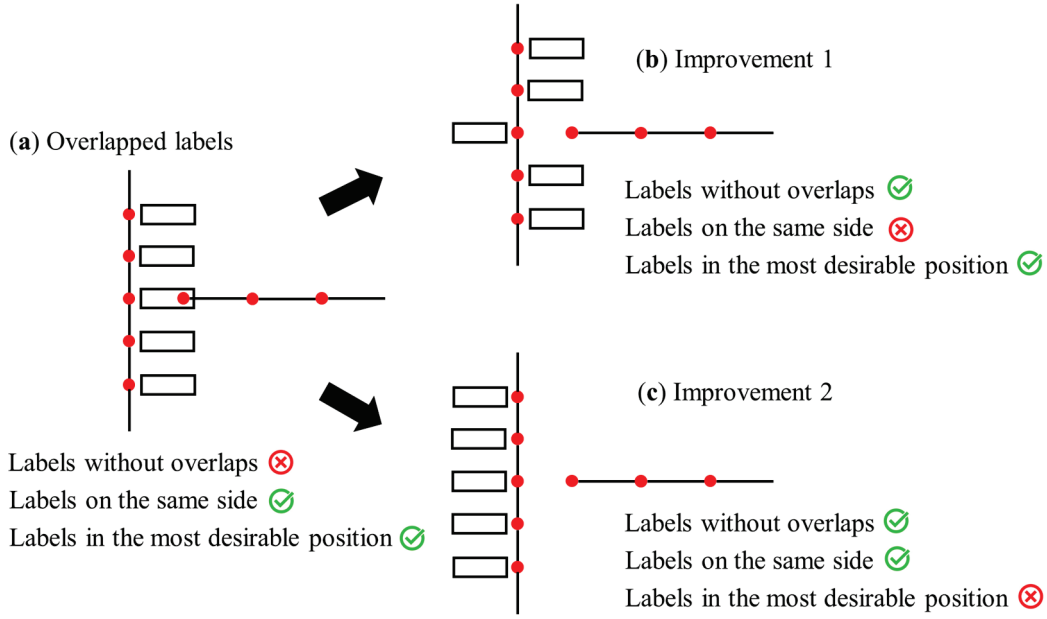


Figure 8. Prioritization among labeling principles exists in schematic metro map design.

2.2. Modeling of Labeling Natural Intelligence

To ensure the labeling natural intelligence to be learned by the ANN model, it is necessary to model them mathematically. As the basic unit to be learned is the labels, we first define the candidate labels for station points. Generally, the direction of labels can be horizontal and vertical. For each direction, eight fixed positions are commonly used. As a result, 16 candidate labels are defined for each station point, as shown in Figure 9.

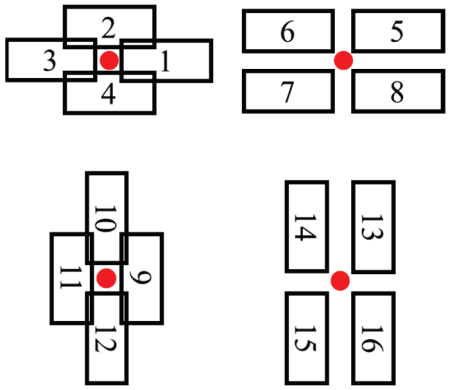
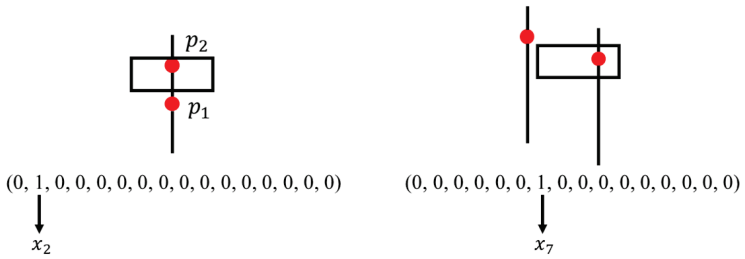


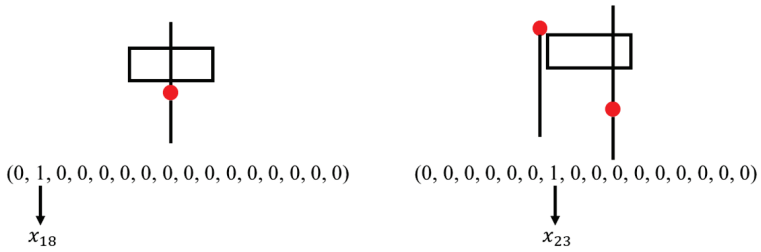
Figure 9. Candidate labels (lower values indicate more preferred positions) for a station point.

2.2.1. Modeling of Overlaps

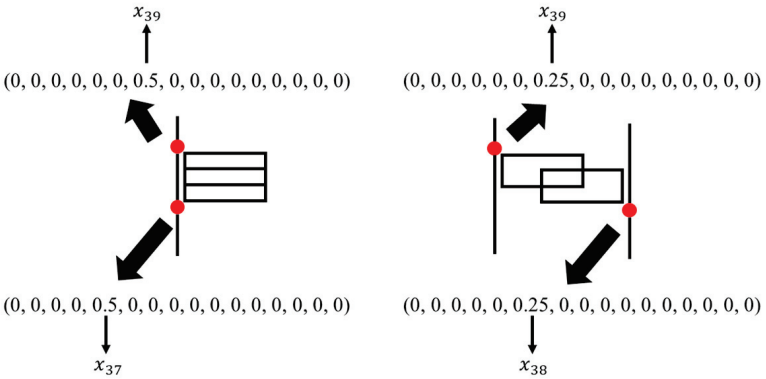
For label–point and label–edge overlaps, we respectively define 16 binary variables ((x_1, x_2, x_3, $\cdots$, x_{16}) and (x_{17}, x_{18}, x_{19}, $\cdots$, x_{32})) to denote whether candidate labels 1 to 16 are overlapped or not. If a candidate label is overlapped, it is recorded as 1, and vice versa as 0. For label and label overlap, we define 16 variables (x_{33}, x_{34}, x_{35}, $\cdots$, x_{48}) to denote the percentage of overlapped area, which ranges from 0 to 1. Some examples are shown in Figure 10. In Figure 10a, the second candidate label of p_1 is overlapped by p_2, hence, the variable x_2 of p_1 is recorded 1.



(**a**) Variables modeling label–point overlap



(**b**) Variables modeling label–edge overlap

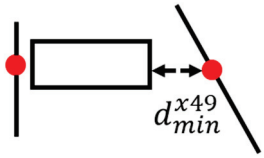


(**c**) Variables modeling label–label overlap

Figure 10. The variables are modeled to express overlapping labels.

2.2.2. Modeling of Ambiguity

The ambiguity of a label can be measured by the distances of the label from the points. The distance between a label and its point is usually fixed in the given metro schematic map. The label is ambiguous when the distances between the label and other points approximate this fixed distance. Therefore, we define 16 variables (x_{49}, x_{50}, x_{51}, $\cdots$, x_{64}) to denote the minimum distances between the candidate labels and other points. It is noted that the distance measure used in this paper is the minimum Euclidean distance between the point and the label contour, as shown in Figure 11.

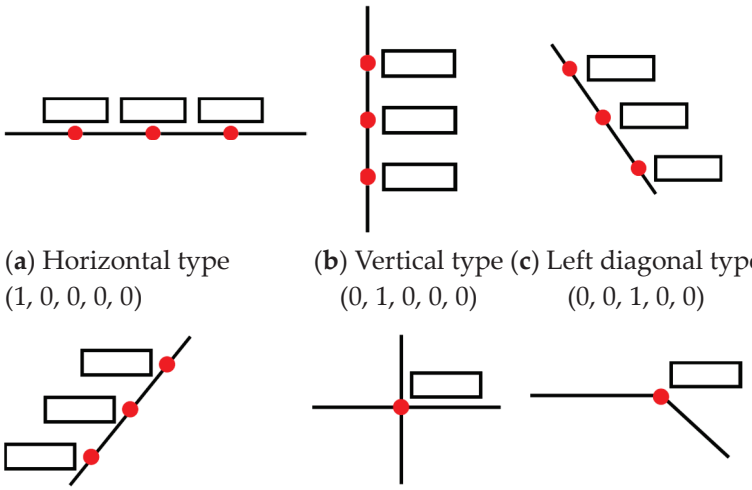


$(d_{min}^{x49}, d_{min}^{x2}, \cdots\cdots, d_{min}^{x64})$

Figure 11. The variables are modelled to express the ambiguity of labels.

2.2.3. Modeling of Passing Line Direction

To model the principle "labels for points touching the same passing line are placed on the same side", the identification of the type of passing line is required. In this paper, we adopt the previous method [7], which employs direction to classify passing lines into five types: horizontal, vertical, left diagonal, right diagonal, and hybrid types. Then, a five-dimensional binary variable (x_{65}, x_{66}, $\cdots$, x_{69}) is used to represent these five passing line directions, as shown in Figure 12.



(**a**) Horizontal type (1, 0, 0, 0, 0)
(**b**) Vertical type (0, 1, 0, 0, 0)
(**c**) Left diagonal type (0, 0, 1, 0, 0)
(**d**) Right diagonal type (0, 0, 0, 1, 0)
(**e**) Hybrid type 1 (0, 0, 0, 0, 1)
(**f**) Hybrid type 2 (0, 0, 0, 0, 1)

Figure 12. Passing lines with various directions.

As the official schematic metro maps are well-designed productions by cartographic experts, they are already considered the most preferred position and the priority among labeling principles. Therefore, this labeling natural intelligence will be observed by learning the relations between label positions and map features. In summary, the mathematical model of labeling natural intelligence for each station point's 16 candidate labels can be defined as the following matrix ($N \times 69$, where N is the number of points, see Table 2), which is used as the input to the ANN model.

Table 2. Characteristic matrix modeling labeling natural intelligence.

	Variables Modeling Label–Point Overlap	Variables Modeling Label–Edge Overlap	Variables Modeling Label–Label Overlap	Variables Modeling Ambiguity of Labels	Variables Modeling Direction of Passing Lines
	$x_1, x_2, x_3, \cdots, x_{16}$	$x_{17}, x_{18}, x_{19}, \cdots, x_{32}$	$x_{33}, x_{34}, x_{35}, \cdots, x_{48}$	$x_{49}, x_{50}, x_{51}, \cdots, x_{64}$	$x_{65}, x_{66}, \cdots, x_{69}$
Point 1	$x_1^1, x_2^1, x_3^1, \cdots, x_{16}^1$	$x_{17}^1, x_{18}^1, x_{19}^1, \cdots, x_{32}^1$	$x_{33}^1, x_{34}^1, x_{35}^1, \cdots, x_{48}^1$	$x_{49}^1, x_{50}^1, x_{51}^1, \cdots, x_{64}^1$	$x_{65}^1, x_{66}^1, \cdots, x_{69}^1$
Point 2	$x_1^2, x_2^2, x_3^2, \cdots, x_{16}^2$	$x_{17}^2, x_{18}^2, x_{19}^2, \cdots, x_{32}^2$	$x_{33}^2, x_{34}^2, x_{35}^2, \cdots, x_{48}^2$	$x_{49}^2, x_{50}^2, x_{51}^2, \cdots, x_{64}^2$	$x_{65}^2, x_{66}^2, \cdots, x_{69}^2$
Point 3	$x_1^3, x_2^3, x_3^3, \cdots, x_{16}^3$	$x_{17}^3, x_{18}^3, x_{19}^3, \cdots, x_{32}^3$	$x_{33}^3, x_{34}^3, x_{35}^3, \cdots, x_{48}^3$	$x_{49}^3, x_{50}^3, x_{51}^3, \cdots, x_{64}^3$	$x_{65}^3, x_{66}^3, \cdots, x_{69}^3$
⋮	⋮	⋮	⋮	⋮	⋮
Point N	$x_1^n, x_2^n, x_3^n, \cdots, x_{16}^n$	$x_{17}^n, x_{18}^n, x_{19}^n, \cdots, x_{32}^n$	$x_{33}^n, x_{34}^n, x_{35}^n, \cdots, x_{48}^n$	$x_{49}^n, x_{50}^n, x_{51}^n, \cdots, x_{64}^n$	$x_{65}^n, x_{66}^n, \cdots, x_{69}^n$

3. ANN-Based Label Placement Method with Natural Intelligence

In this section, we propose an ANN-based label placement method. To implement this method, the construction of the ANN model is required. Then, the training and testing datasets are developed.

3.1. Construction of the ANN Labeling Model

As mentioned in the previous section, ANN has a good generalization capability to perform well on unseen or new data after being trained on a dataset [31–33]. Several common architectures or models of ANN (e.g., multilayer feedforward, MLF, and convolutional neural network, CNN) have been developed over the years to address various tasks and challenges, such as image and speech recognition [34,35], natural language processing [36], and recommendation systems [37]. In cartography, ANN can be applied to enhance various aspects, such as map generalization [38–41] and map production [42,43].

In this study, we treat the label placement as a supervised learning task. Then, the MLF model is selected for its suitability for supervised tasks [44]. A complete MLF model typically contains the following components: input, hidden, and output layers. The number of hidden layers and the number of neurons in the hidden layers are determined by a "trial-by-error" method [7]. This method involves initially training neural networks with varying numbers of hidden neurons, followed by an assessment based on generalization error to determine the optimal number. For activation functions, preference is given to nonlinear options. Among these, the Rectified Linear Unit (ReLU) is employed in this study for its computational efficiency and nonlinear properties. Finally, we construct the ANN model using PyTorch, which is a popular open-source machine learning library developed by Facebook's AI Research lab (FAIR). Figure 13 shows the constructed ANN model.

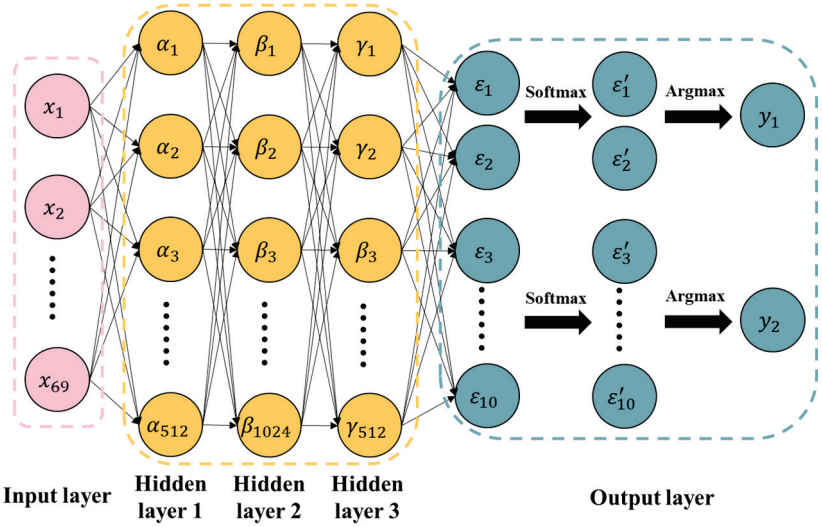


Figure 13. The constructed ANN model in this study.

The model's initial values of parameters and hyperparameters are required before the training and testing phase. Usually, the parameters are randomly initialized and dynamically optimized in the training process. The hyperparameters (e.g., learning rate) are set manually, and their initial values are listed in Table 3.

Table 3. Initial values of parameters.

Hyperparameters	Initial Values	Explanation
Learning rate	0.005	Controls the balance between convergence speed and stability
Batch size	5	The number of training examples utilized in one iteration of the training process
Momentum	0.9	Accelerates optimization, smooths gradient updates, and helps overcome noisy gradients and oscillations
Step size Gamma	7 0.1	Affects the change in the learning rate
Number of epochs	50	Determines how many times the network learns from the entire training dataset

3.2. Training and Testing the ANN Labeling Model

The original training and testing datasets are collected as pictures from the official schematic metro maps, which are well-designed productions by cartographic experts. To fill the characteristic matrix ($N \times 69$), we vectorize pictures of metro maps by ArcGIS 10.6.1 software. It is noted that the sizes of vector pictures are various, leading to significant differences in the size of the labels. To handle this, we keep the label size similar, i.e., the same size of language characters in name labels. Specifically, the lengths of characters on schematic maps are utilized as benchmarks to guide the "zoom in" or "zoom out" function of these vectorized pictures. That is, the side lengths of the minimum bounding rectangle (MBR) of the characters on different schematic metro maps are equal. This standardization is achieved by dividing all coordinates of the points on a schematic metro map by the length of the MBR of a character. Then, we generate the candidate labels for each station point, as shown in Figure 14. Finally, we fill the characteristic matrix with ArcGIS functions, such as "Select Layer by Location" and "Near".

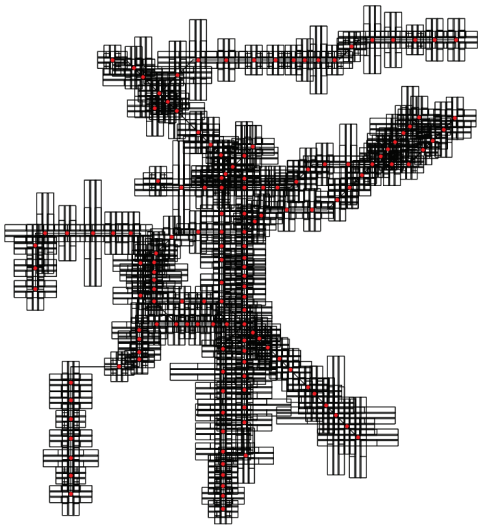


Figure 14. The candidate labels for each station point.

It is noted that the common metric "accuracy" is limited when evaluating the testing labeling outcomes because the same input may produce multiple correct outputs. This may be attributed to the individual preferences of schematic network map designers [7].

4. Experimental Evaluation

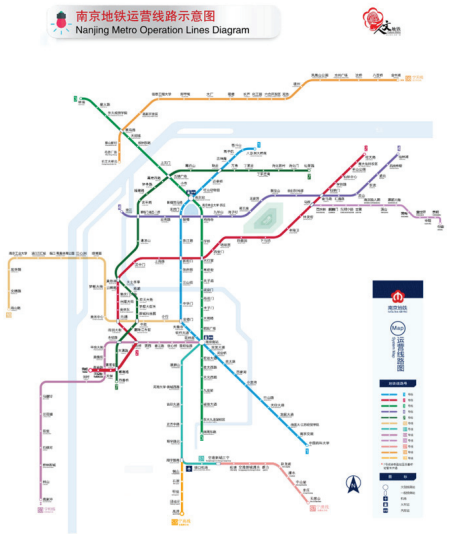
Quantitative and qualitative evaluations have been conducted in this section. Five evaluation metrics of the labeling results have been compared between our method and the benchmark [7]. On the other hand, a survey has been conducted by asking users to select the more pleasing labeling results.

4.1. Experimental Data

The official schematic metro maps were collected from 15 cities (Chengdu, Guangzhou, Hong Kong, Nanjing, Shenzhen, Suzhou, Tianjin, Wuhan, Beijing, Xiamen, Chongqing, Singapore, Madrid, New York, and London) in different countries, as shown in Figure 15. A total of 3048 station points are sampled from these maps. The training set is from 12 cities (Chengdu, Guangzhou, Hong Kong, Shenzhen, Tianjin, Wuhan, Beijing, Xiamen, Chongqing, Madrid, New York, and London) with 2635 station points. The test set is from three cities (Nanjing, Suzhou, and Singapore) with 413 stations.



Tianjin



Nanjing

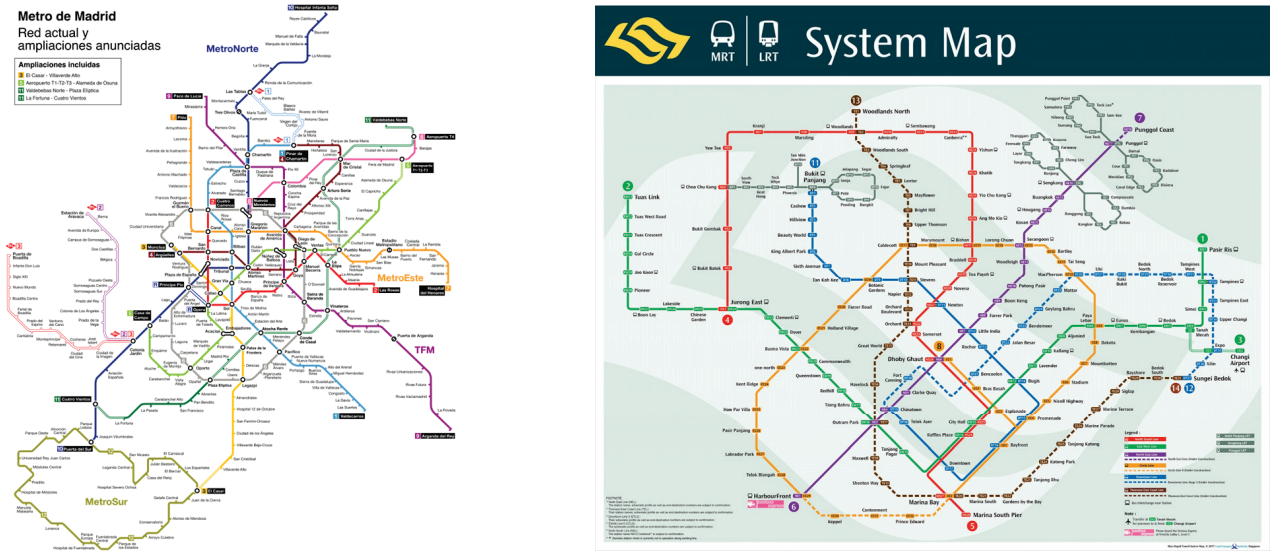


Guangzhou



Shanghai

Figure 15. *Cont.*



Madrid Singapore

Figure 15. Schematic metro maps of cities (some station names are presented in Chinese characters).

4.2. Experimental Evaluation Method

Four metrics corresponding to the most common labeling natural intelligence (see Section 2) are used for quantitative evaluation, i.e., the number of overlapping labels ($Num_{\text{overlapping}}$), the sum of the closest distance from each label to other points (Dis_{minimum}), the number of labels placed at the most preferred position ($Num_{\text{preference}}$), and the number of labels placed on the same side ($Num_{\text{consistency}}$). On the other hand, the map-load level (MLL) is a crucial aspect of map design, and it was used to evaluate schematic metro maps.

$Num_{\text{overlapping}}$ includes overlap between label and point ($Overlap_{\text{label-point}}$), overlap between label and edge ($Overlap_{\text{label-edge}}$), as well as overlap between label and label ($Overlap_{\text{label-label}}$). MLL can be calculated by the Graphic Map Load Measuring Tool (GMLMT) [45]. In order to calculate the $Num_{\text{consistency}}$, we define the side where the labels should be placed, which is determined by the majority.

Qualitative evaluation asks users to select the more pleasing labeling results from our method and benchmark. In this survey, we obtained the selection of 42 participants through the "Wenjuanxing" online platform.

4.3. Experimental Results

Figures 16–18 show the labeling results of our method and the benchmark. Table 4 lists the $Num_{\text{overlapping}}$, $Overlap_{\text{label-point}}$, $Overlap_{\text{label-edge}}$, and $Overlap_{\text{label-label}}$. The results show that the $Num_{\text{overlapping}}$ of our method is significantly lower than that of the benchmark. More precisely, the $Overlap_{\text{label-point}}$, $Overlap_{\text{label-edge}}$, and $Overlap_{\text{label-label}}$ of our method are lower than those of the benchmark, accounting for 0.00%, 4.12%, and 20.58% of the total number of labels (413), compared with 4.17%, 14.29%, and 35.11% for the benchmark, respectively. These results indicate that our method is effective in avoiding overlapping labels.

Table 5 shows the Dis_{minimum} of our method and the benchmark. The results show that the Dis_{minimum} of our method is larger than that of the benchmark. This indicates that our method is good at avoiding ambiguous labels, as shown in Figure 19.

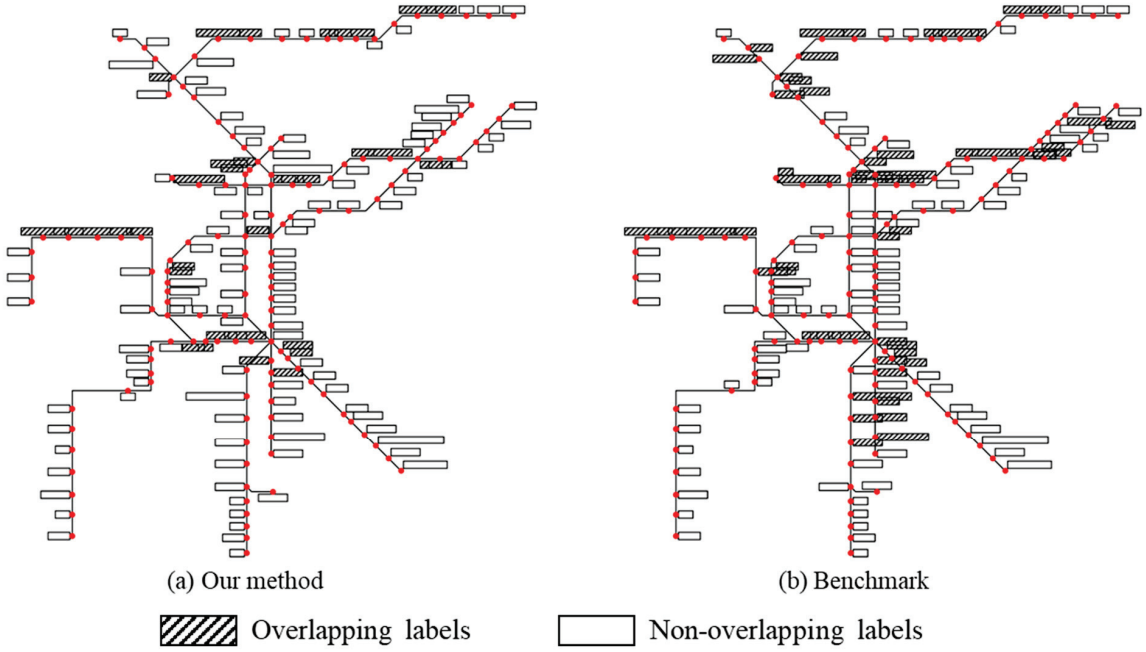


Figure 16. Nanjing schematic metro maps are produced using our method and the benchmark.

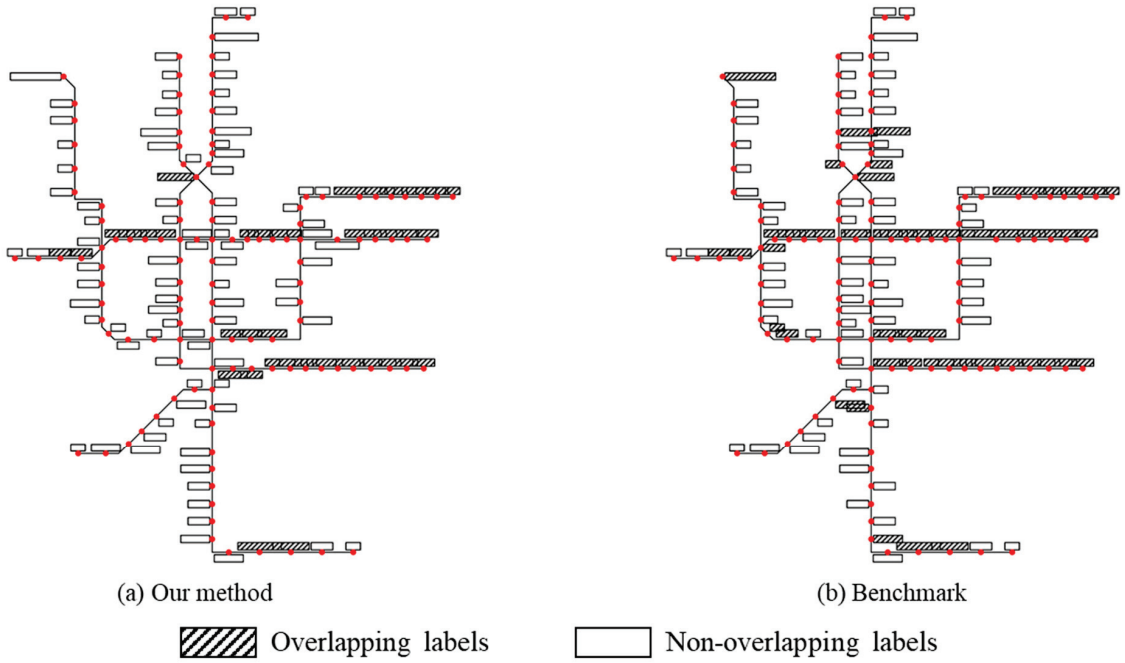


Figure 17. Suzhou schematic metro maps are produced using our method and the benchmark.

Table 4. The number of overlapping labels produced by our method and the benchmark.

City	Method	$Overlap_{label-point}$	$Overlap_{label-edge}$	$Overlap_{label-label}$	$Num_{overlapping}$
Nanjing	Our method	0	6	33	39
	Benchmark	9	24	58	91
Suzhou	Our method	0	2	39	41
	Benchmark	1	7	54	62
Singapore	Our method	0	9	12	21
	Benchmark	7	28	33	68

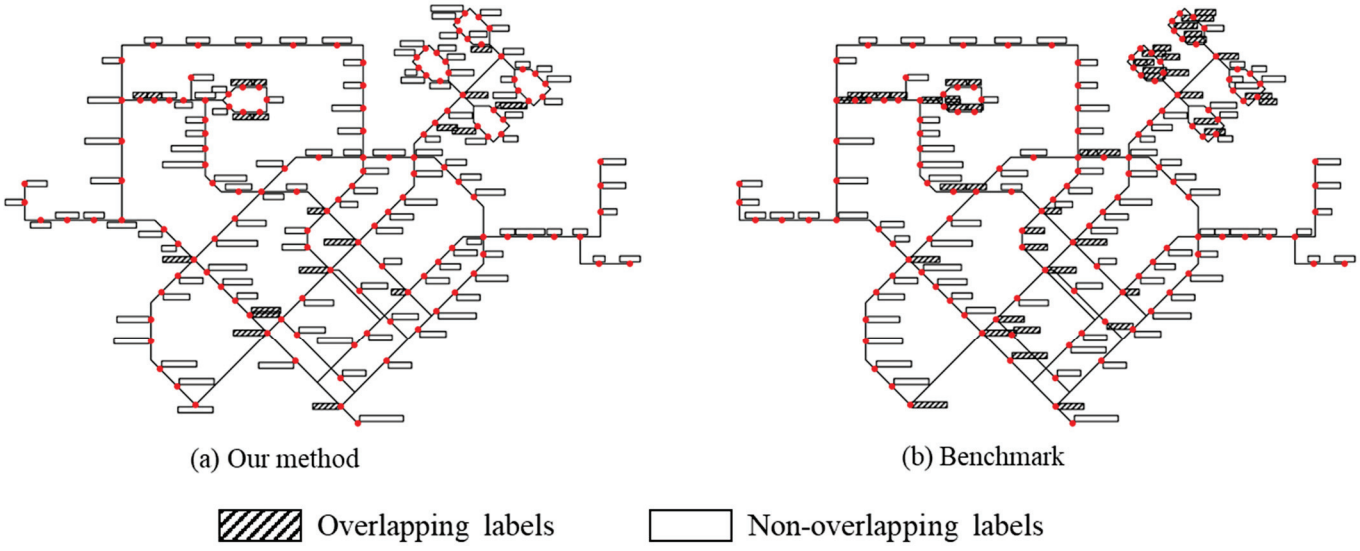


Figure 18. Singapore schematic metro maps are produced using our method and the benchmark.

Table 5. The Dis_{minimum} of our method and the benchmark.

City	Method	Dis_{minimum}
Nanjing	Our method	2234.7
	Benchmark	1807.3
Suzhou	Our method	1939.7
	Benchmark	1792.6
Singapore	Our method	3542.1
	Benchmark	3111.6

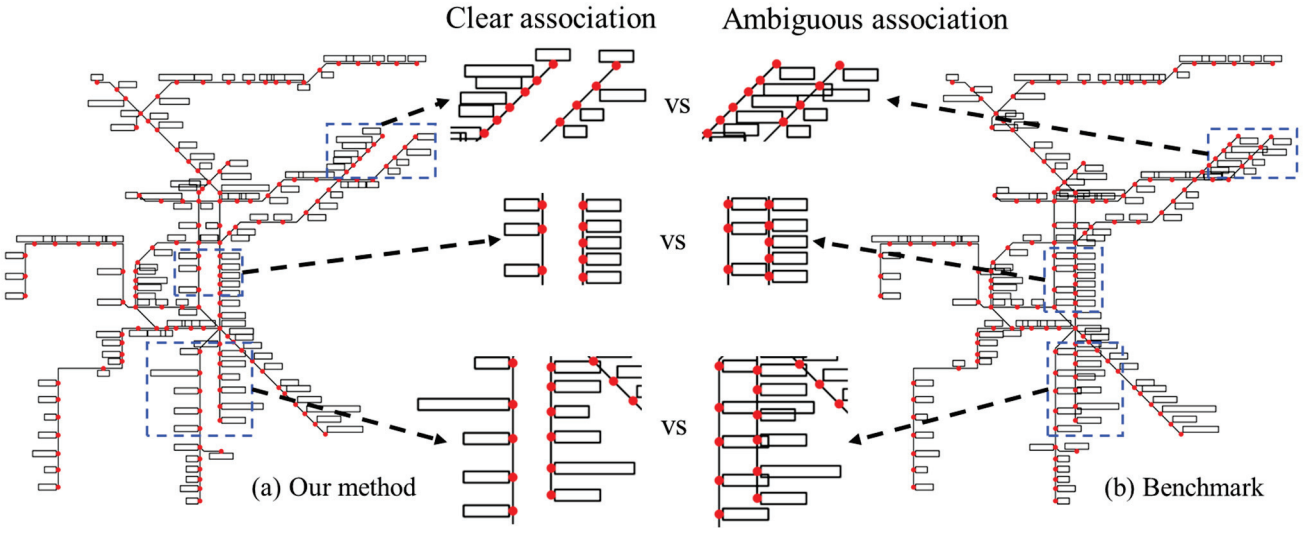


Figure 19. Comparison of ambiguous labels between our method and the benchmark.

Table 6 lists the $Num_{\text{preference}}$ and $Num_{\text{consistency}}$ of our method and the benchmark. It was found that the $Num_{\text{preference}}$ of our method is lower than that of the benchmark. This is because our method prioritizes the avoidance of overlapping labels at the expense of preference. On the other hand, $Num_{\text{consistency}}$ is comparable between our method and the benchmark.

Table 6. The $Num_{\text{preference}}$ and $Num_{\text{consistency}}$ of our method and the benchmark.

City	Method	$Num_{\text{preference}}$	$Num_{\text{consistency}}$
Nanjing	Our method	104	135
	Benchmark	138	136
Suzhou	Our method	89	111
	Benchmark	120	116
Singapore	Our method	121	129
	Benchmark	135	125

Figures 20–22 show the map-load level (ranging from 0 to 100) distribution on the resulting maps of our method and the benchmark. Specifically, the maps are divided into 100 grids, and the map-load level for each grid is calculated. We focus on the grids that contain overlapping labels (marked by green boxes) or ambiguous labels (marked by blue boxes) in the benchmark but do not in our method.

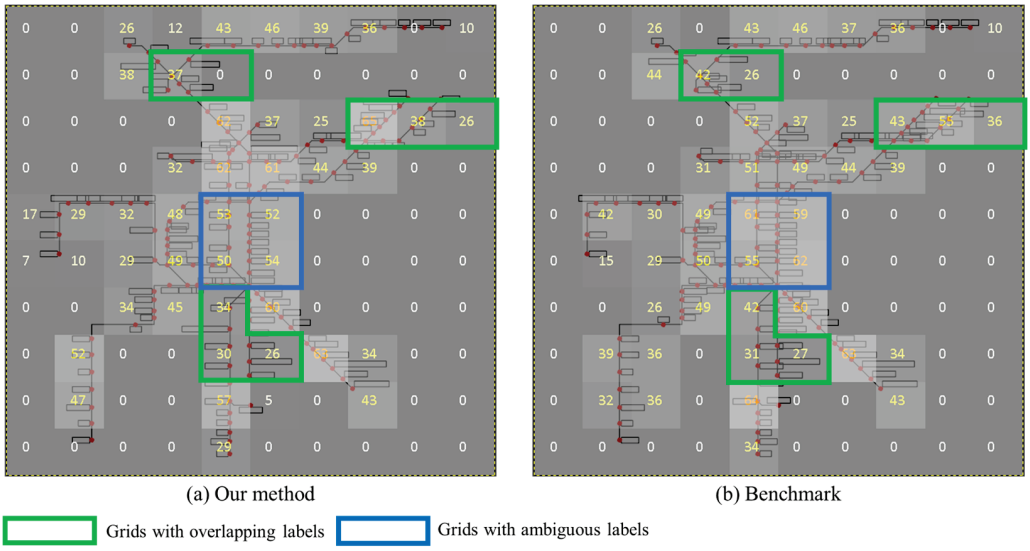


Figure 20. Map-load level distribution of Nanjing schematic metro maps.

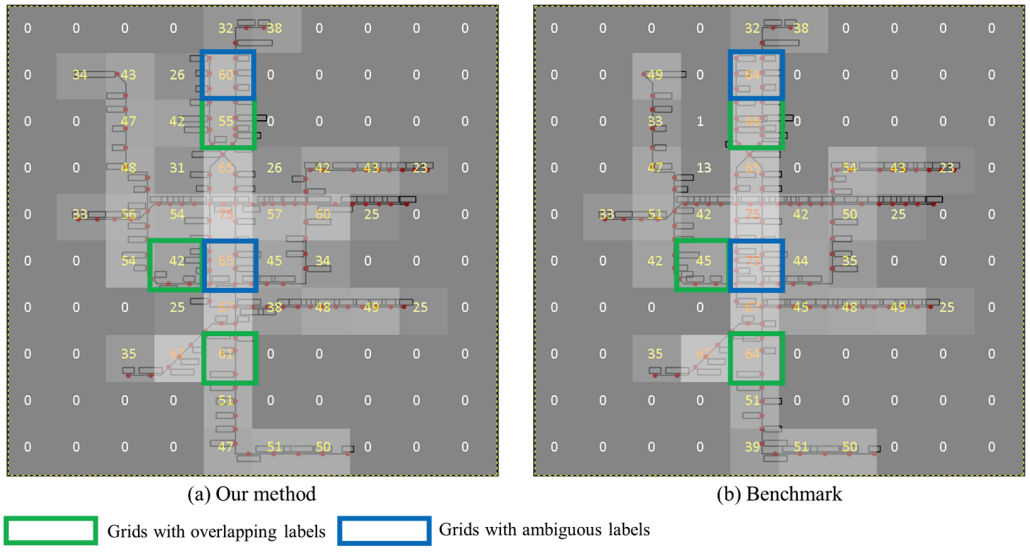


Figure 21. Map-load level distribution of Suzhou schematic metro maps.

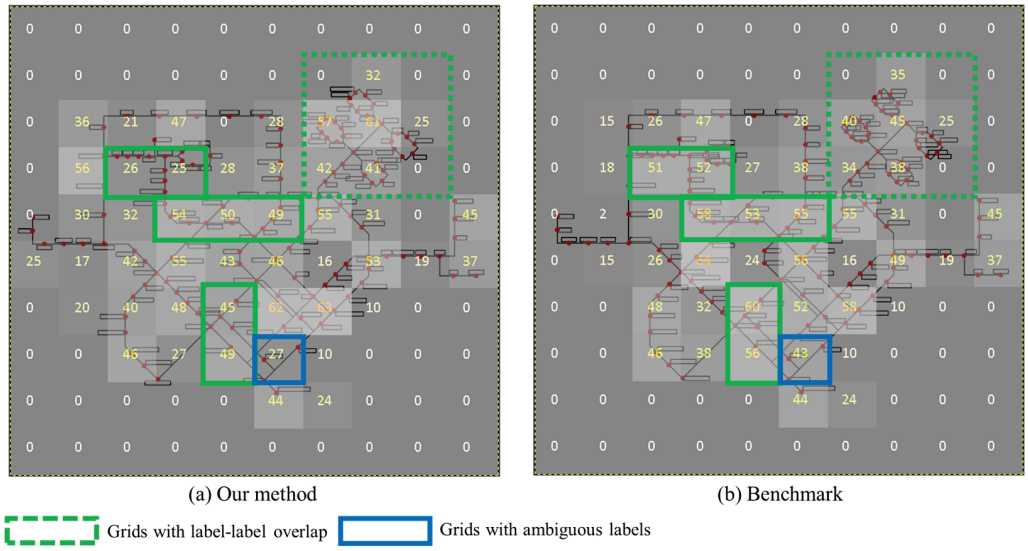


Figure 22. Map-load level distribution of Singapore schematic metro maps.

It was found that the map-load levels of the blue boxes and most of the green boxes decreased in our resulting maps. This finding indicates that our method can place labels with a lower map-load level. Further, we found that the map-load levels decrease after eliminating label–label overlaps (green solid boxes) but increase after eliminating label–point or label–edge overlaps (green dotted boxes, see Figure 22). This is because the visual complexity of the label–point or label–edge overlap seems lower than that of the label–label overlap. However, it is clear that our label placements in green dotted boxes, despite having larger map-load levels, are more readable. This implies that the map-load level calculated by GMLMT is only an objective estimation of visual complexity. Therefore, a subjective evaluation of readability for label placement is required.

Table 7 presents the preferences of 42 participants regarding label placement generated by our method compared to the benchmark. Results show that 35 participants (83.3%), 30 participants (71.4%), and 31 participants (73.8%) preferred our labeling results for Nanjing, Suzhou, and Singapore, respectively. These results reveal that the label placements of our method are more pleasing than the benchmark.

Table 7. Preferences of schematic metro maps between our method and the benchmark.

City	Method	Number	Proportion
Nanjing	Our method	35	83.3%
	Benchmark	7	16.7%
Suzhou	Our method	30	71.4%
	Benchmark	12	28.6%
Singapore	Our method	31	73.8%
	Benchmark	11	26.2%

5. Discussion

5.1. Impact Analysis of Different Characteristics and Settings

An ablation study has been conducted to analyze the impact of different components or settings of our method on the resulting schematic metro maps. Specifically, hidden layers-based and characteristic variables-based ablation experiments have been performed, respectively (see Table 8 for details on these experiments).

Table 9 shows the results of hidden layers-based ablation experiments. No significant impact was found after removing any hidden layers. Table 10 shows the results of characteristic variables-based ablation experiments. It was found that $Overlap_{\text{label-point}}$,

$Overlap_{\text{label-edge}}$, and $Overlap_{\text{label-label}}$ significantly increase while Dis_{minimum} decreases after removing characteristic variable 1 (label–point overlaps) and characteristic variable 2 (label–edge overlaps). These results indicate that these two characteristics play an important role in avoiding overlapping and ambiguous labels. On the other hand, this implies that the remaining characteristics may be redundant in the original experiment.

Table 8. Hidden layers-based and characteristic variables-based ablation experiments.

	Experiment	Explanation
Hidden layers-based	*Remove_H*1	Remove hidden layer 1 (69 × 512)
	*Remove_H*2	Remove hidden layer 2 (512 × 1024)
	*Remove_H*3	Remove hidden layer 3 (1024 × 512)
Characteristic variables-based	*Remove_C*1	Remove variables of label–point overlap
	*Remove_C*2	Remove variables of label–edge overlap
	*Remove_C*3	Remove variables of label–label overlap
	*Remove_C*4	Remove variables of ambiguity of labels
	*Remove_C*5	Remove variables of direction of passing lines

Table 9. Results of hidden layers-based ablation experiments.

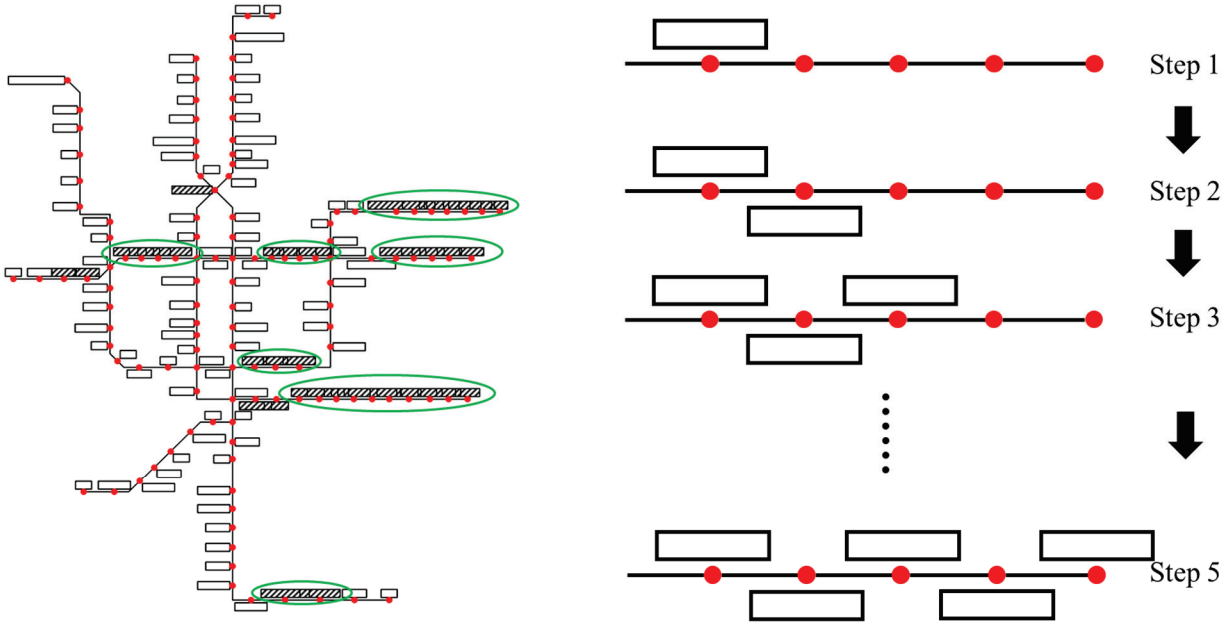
City	Experiment	$Overlap_{\text{label-point}}$	$Overlap_{\text{label-edge}}$	$Overlap_{\text{label-label}}$	$Num_{\text{overlapping}}$	Dis_{minimum}
Nanjing	*Remove_H*1	0	6	32	38	2178.9
	*Remove_H*2	1	6	37	44	2207.1
	*Remove_H*3	0	7	35	42	2252.9
	Reference	0	6	33	39	2234.7
Suzhou	*Remove_H*1	0	2	37	39	1961.3
	*Remove_H*2	0	2	40	42	1958.5
	*Remove_H*3	0	2	41	43	2001.7
	Reference	0	2	39	41	1939.7
Singapore	*Remove_H*1	0	9	14	23	3531.4
	*Remove_H*2	0	9	14	23	3474.7
	*Remove_H*3	0	8	8	16	3550.7
	Reference	0	9	12	21	3542.1

Table 10. Results of characteristic variables-based ablation experiments.

City	Experiment	$Overlap_{\text{label-point}}$	$Overlap_{\text{label-edge}}$	$Overlap_{\text{label-label}}$	$Num_{\text{overlapping}}$	Dis_{minimum}
Nanjing	*Remove_C*1	5	53	39	97	1973.2
	*Remove_C*2	4	15	40	59	2131.2
	*Remove_C*3	0	6	37	43	2151.2
	*Remove_C*4	0	5	41	46	2158.2
	*Remove_C*5	0	7	33	40	2207.0
	Reference	0	6	33	39	2234.7
Suzhou	*Remove_C*1	3	16	44	63	1894.3
	*Remove_C*2	0	4	42	46	1948.0
	*Remove_C*3	0	2	41	43	1939.8
	*Remove_C*4	0	2	39	41	1927.8
	*Remove_C*5	0	2	40	42	1957.3
	Reference	0	2	39	41	1939.7
Singapore	*Remove_C*1	3	76	22	101	2960.0
	*Remove_C*2	5	23	27	55	3354.9
	*Remove_C*3	0	9	16	25	3483.5
	*Remove_C*4	0	8	10	18	3516.9
	*Remove_C*5	0	8	12	20	3497.8
	Reference	0	9	12	21	3542.1

5.2. Reduce Overlapping Labels Using an Optimization Algorithm

The overlapping labels still exist in our labeling results, and most of them are caused by adjacent station points touching a horizontal passing line, as shown in Figure 23a. On official schematic metro maps, labels of such points are usually progressively placed crosswise up and down based on heuristic information, as shown in Figure 23b. However, such heuristic information heavily depends on human intelligence, and it is difficult to integrate them into the characteristic matrix.



(a) Overlaps caused by adjacent station points touching a horizontal passing line

(b) Labels are progressively placed crosswise up and down based on heuristic information

Figure 23. Major overlaps of our results and the manual placement method.

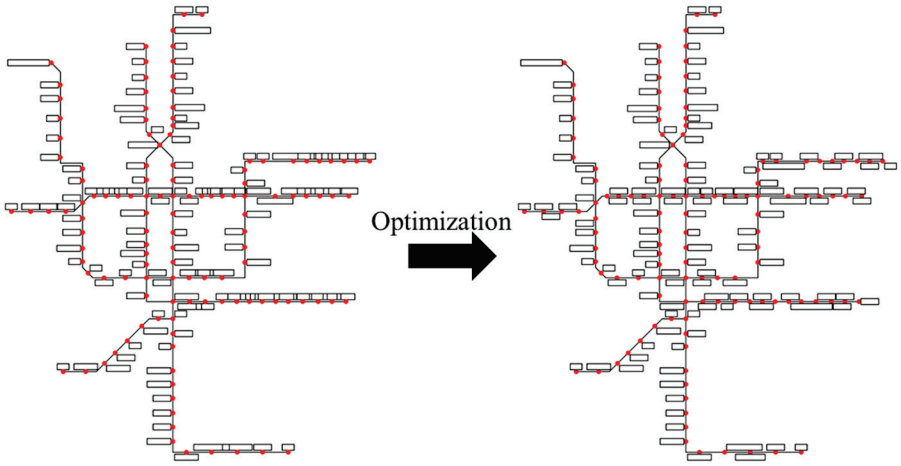
The traditional method was employed to reduce these remaining overlapping labels automatically. Specifically, the label placement is treated as a combinatorial optimization problem in a limited cartographic space. To solve the optimization problem means to maximize or minimize one or more objective functions.

In this process, an initial label placement is required to input, and a random placement is commonly used due to simplicity. However, such placement is inappropriate for schematic metro maps because these maps require special constraints, e.g., labels from the same passing line should be placed on the same side as much as possible.

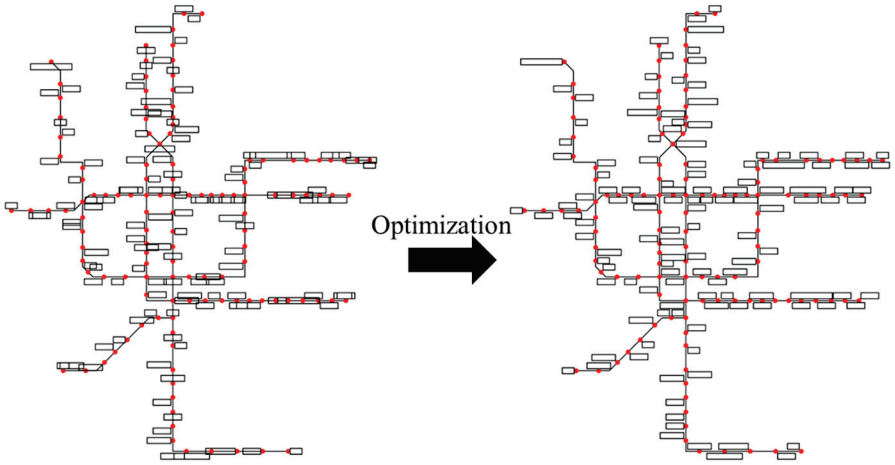
The labeling result of our method can be used as an appropriate initial label placement. In order to verify this idea, we employed a genetic algorithm (one commonly used optimization algorithm) to optimize the following objective function:

$$Objective\ function = Min\left(Overlap_{\text{label}-\text{point}} + Overlap_{\text{label}-\text{edge}} + Overlap_{\text{label}-\text{label}}\right) \quad (1)$$

Figure 24 shows the results optimized from our label placement and the random label placement. No overlapping labels appear in both optimized results. It is clear that the results from our label placement exhibit better preference and consistency.



(a) Result (right) optimized from the initial label placement of our method (left)



(b) Result (right) optimized from the initial label placement of the random method (left)

Figure 24. The non-overlapping results were optimized from two initial label placements.

On the other hand, Table 11 shows the overlapping labels of two initial label placements and the iterations required for optimization. The findings suggest that the optimization from our label placement performs fewer iterations, implying its effectiveness for practical application.

Table 11. The number of overlapping labels and the number of iterations performed for two initial label placements.

Initial Placement	Overlapping Labels	Number of Iterations
Our method	41	18
Random method	105	58

6. Conclusions

On schematic metro maps, high-quality name labels are helpful to passengers performing metro route planning and orientation tasks. It has been reported that an artificial neural network (ANN) is an effective and efficient way to place high-quality labels. In this study, we developed an improved labeling method by integrating relations between label positions and both connected and surrounding features into the ANN model.

Quantitative and qualitative evaluations with a benchmark have been conducted using three official schematic metro maps. Quantitative results show our method can produce

schematic metro maps with fewer overlapping labels than the benchmark. Specifically, the number of three types of overlaps (label–point, label–edge, and label–label) by our method is lower than that of the benchmark, accounting for 0.00%, 4.12%, and 20.58% of the total number of labels, compared with 4.17%, 14.29%, and 35.11% for the benchmark, respectively. On the other hand, our method is good at avoiding ambiguous labels and also trying to ensure labels from the same passing line on the same side. It should be noted that the reduction in overlapping and ambiguous labels is achieved at the cost of the label position preference. Qualitative results show that most users (around 75%) are more satisfied with our schematic maps than those of the benchmark. Thus, based on the experimental results, it may be concluded that the improved ANN-based label placement method is effective.

On the other hand, our method can be extended to place labels for topographic maps, which usually include multiple features (i.e., points, lines, and areas). The key to developing an ANN-based method for labeling topographic maps is to define effective relationships between labels and features based on labeling natural intelligence. It is believed that this method can significantly improve the quality and efficiency of label placement on topographic maps.

There are limitations to this new method. First, a particular type of overlapping label (i.e., occurring at adjacent station points touching a horizontal passing line) still exists in our labeling results. Although an optimization algorithm-based method has been discussed to address this limitation, future work will focus on how to model it to achieve completely automatic label placement based on artificial neural networks. Second, the developed characteristic matrix may include redundant variables, and an improved matrix will be explored.

Author Contributions: Conceptualization, Zhiwei Wu; funding acquisition, Tian Lan; investigation, Zhiwei Wu, Chenzhen Sun, Donglin Cheng, Xing Shi, and Guangjun Zeng; methodology, Zhiwei Wu and Tian Lan; supervision, Tian Lan; visualization, Meisheng Chen; writing—original draft, Zhiwei Wu; writing—review and editing, Zhiwei Wu and Tian Lan. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant Nos. 42101442 and 42394063).

Data Availability Statement: The data presented in this study are available upon request from the corresponding authors. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nollenburg, M.; Wolff, A. Drawing and labeling high-quality metro maps by mixed-integer programming. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 626–641. [CrossRef] [PubMed]
2. Roberts, M.J.; Gray, H.; Lesnik, J. Preference versus performance: Investigating the dissociation between objective measures and subjective ratings of usability for schematic metro maps and intuitive theories of design. *Int. J. Hum. Comput. Stud.* **2017**, *98*, 109–128. [CrossRef]
3. Morrison, A. Public transport maps in western European cities. *Cartogr. J.* **1996**, *33*, 93–110. [CrossRef]
4. Hong, S.-H.; Merrick, D.; do Nascimento, H.A.D. The metro map layout problem. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 482–491.
5. Nöllenburg, M. A survey on automated metro map layout methods. In Proceedings of the 1st Schematic Mapping Workshop, Essex, UK, 2–3 April 2014.
6. Lan, T.; Li, Z.L.; Peng, Q.; Gong, X.Y. Automated labeling of schematic maps by optimization with knowledge acquired from existing maps. *Trans. GIS* **2020**, *24*, 1722–1739. [CrossRef]
7. Lan, T.; Li, Z.L.; Wang, J.C.; Gong, C.Y.; Ti, P. An ANNs-based method for automated labelling of schematic metro maps. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 36. [CrossRef]
8. Avelar, S.; Hurni, L. On the design of schematic transport maps. *Cartogr. Int. J. Geogr. Inf. Geovis.* **2006**, *41*, 217–228. [CrossRef]
9. Stott, J.M.; Rodgers, P. Metro map layout using multicriteria optimization. In Proceedings of the Eighth International Conference on Information Visualisation, London, UK, 16 July 2004; pp. 355–362.

10. Avelar, S. Convergence analysis and quality criteria for an iterative schematization of networks. *GeoInformatica* **2007**, *11*, 497–513. [CrossRef]
11. Stott, J.; Rodgers, P.; Martínez-Ovando, J.C.; Walker, S.G. Automatic metro map layout using multicriteria optimization. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 101–114. [CrossRef] [PubMed]
12. Wang, Y.S.; Chi, M.T. Focus+context metro maps. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 2528–2535. [CrossRef]
13. Roberts, M.J.; Newton, E.J.; Lagattolla, F.D.; Hughes, S.; Hasler, M.C. Objective versus subjective measures of Paris Metro map usability: Investigating traditional octolinear versus all-curves schematics. *Int. J. Hum.-Comput. Stud.* **2013**, *71*, 363–386. [CrossRef]
14. Mark Ware, J.; Taylor, G.E.; Anand, S.; Thomas, N. Automated production of schematic maps for mobile applications. *Trans. GIS* **2006**, *10*, 25–42. [CrossRef]
15. Li, Z.L.; Dong, W.H. A stroke-based method for automated generation of schematic network maps. *Int. J. Geogr. Inf. Sci.* **2010**, *24*, 1631–1647. [CrossRef]
16. Ware, M.; Richards, N. An ant colony system algorithm for automatically schematizing transport network data sets. In Proceedings of the IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; pp. 1892–1900.
17. Ti, P.; Li, Z.L. Generation of schematic network maps with automated detection and enlargement of congested areas. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 521–540. [CrossRef]
18. Lan, T.; Li, Z.L.; Ti, P. Integrating general principles into mixed-integer programming to optimize schematic network maps. *Int. J. Geogr. Inf. Sci.* **2019**, *33*, 2305–2333. [CrossRef]
19. Schwetman, J.D. Harry Beck's London Underground map: A convex lens for the global city. *Transfers* **2014**, *4*, 86–103. [CrossRef]
20. Imhof, E. Positioning names on maps. *Am. Cartogr.* **1975**, *2*, 128–144. [CrossRef]
21. Yoeli, P. The logic of automated map lettering. *Cartogr. J.* **1972**, *9*, 99–108. [CrossRef]
22. Christensen, J.; Marks, J.; Shieber, S. An empirical study of algorithms for point-feature label placement. *ACM Trans. Graph.* **1995**, *14*, 203–232. [CrossRef]
23. Oucheikh, R.; Harrie, L. A feasibility study of applying generative deep learning models for map labeling. *Cartogr. Geogr. Inf. Sci.* **2024**, *51*, 168–191. [CrossRef]
24. Wu, H.-Y.; Niedermann, B.; Takahashi, S.; Roberts, M.J.; Nöllenburg, M. A survey on transit map layout—From design, machine, and human perspectives. *Comput. Graph. Forum* **2020**, *39*, 619–646. [CrossRef]
25. Ribeiro, J.T.; Rijo, R.; Leal, A. Fast automatic schematics for public transport spider maps. *Procedia Technol.* **2012**, *5*, 659–669. [CrossRef]
26. Oke, O.; Siddiqui, S. Efficient automated schematic map drawing using multiobjective mixed integer programming. *Comput. Oper. Res.* **2015**, *61*, 1–17. [CrossRef]
27. Wolff, A. Drawing subway maps: A survey. *Inform. Forsch. Und Entwickl.* **2007**, *22*, 23–44. [CrossRef]
28. Niedermann, B.; Haunert, J.-H. An algorithmic framework for labeling network maps. *Algorithmica* **2018**, *80*, 1493–1533. [CrossRef]
29. Netzel, R.; Ohlhausen, B.; Kurzhals, K.; Woods, R.; Burch, M.; Weiskopf, D. User performance and reading strategies for metro maps: An eye tracking study. *Spat. Cogn. Comput.* **2017**, *17*, 39–64. [CrossRef]
30. Jacobsen, B.; Wallinger, M.; Kobourov, S.; Nöllenburg, M. MetroSets: Visualizing sets as metro maps. *IEEE Trans. Vis. Comput. Graph.* **2021**, *27*, 1257–1267. [CrossRef]
31. Jiang, Y.N. Prediction of monthly mean daily diffuse solar radiation using artificial neural networks and comparison with other empirical models. *Energy Policy* **2008**, *36*, 3833–3837. [CrossRef]
32. Basheer, I.A.; Hajmeer, M. Artificial neural networks: Fundamentals, computing, design, and application. *J. Microbiol. Methods* **2000**, *43*, 3–31. [CrossRef] [PubMed]
33. Lu, Y.; Ye, L.; Su, Z.Q.; Zhou, L.M.; Cheng, L. Artificial neural network (ANN)-based crack identification in aluminum plates with lamb wave signals. *J. Intell. Mater. Syst. Struct.* **2009**, *20*, 39–49. [CrossRef]
34. Yang, C.C.; Prasher, S.O.; Landry, J.; DiTommaso, A. Application of artificial neural networks in image recognition and classification of crop and weeds. *Can. Agric. Eng.* **2000**, *42*, 147–152.
35. Trentin, E.; Gori, M. A survey of hybrid ANN/HMM models for automatic speech recognition. *Neurocomputing* **2001**, *37*, 91–126. [CrossRef]
36. Rodzin, S.; Bova, V.; Kravchenko, Y.; Rodzina, L. Deep learning techniques for natural language processing. In *Lecture Notes in Networks and Systems*; Springer: Cham, Switzerland, 2022; pp. 121–130.
37. Brintha, N.C.; Nagaraj, P.; Tejasri, A.; Durga, B.V.; Teja, M.T.; Kumar, M.N.V.P. A food recommendation system for predictive diabetic patients using ANN and CNN. In Proceedings of the 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 22–24 June 2022; pp. 1364–1371.
38. Courtial, A.; El Ayedi, A.; Touya, G.; Zhang, X. Exploring the potential of deep learning segmentation for mountain roads generalisation. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 338. [CrossRef]
39. Touya, G.; Zhang, X.; Lokhat, I. Is deep learning the new agent for map generalization? *Int. J. Cartogr.* **2019**, *5*, 142–157. [CrossRef]
40. Feng, Y.; Thiemann, F.; Sester, M. Learning cartographic building generalization with deep convolutional neural networks. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 258. [CrossRef]
41. Zhou, Q.; Li, Z.L. Use of artificial neural networks for selective omission in updating road networks. *Cartogr. J.* **2014**, *51*, 38–51. [CrossRef]

42. Henry, C.J.; Storie, C.D.; Palaniappan, M.; Alhassan, V.; Swamy, M.; Aleshinloye, D.; Curtis, A.; Kim, D. Automated LULC map production using deep neural networks. *Int. J. Remote Sens.* **2019**, *40*, 4416–4440. [CrossRef]
43. Kang, Y.H.; Gao, S.; Roth, R.E. Transferring multiscale map styles using generative adversarial networks. *Int. J. Cartogr.* **2019**, *5*, 115–141. [CrossRef]
44. Ishak, S.; Al-Deek, H. Performance of automatic ANN-based incident detection on freeways. *J. Transp. Eng.* **1999**, *125*, 281–290. [CrossRef]
45. Barvir, R.; Vit, V. Graphic Map Load Measuring Tool—Development and verification. *Int. J. Cartogr.* **2021**, *7*, 285–303. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of
Geo-Information

MDPI

Article

Road Network Intelligent Selection Method Based on Heterogeneous Graph Attention Neural Network

Haohua Zheng [1,2,3], Jianchen Zhang [1,2,3,4,*], Heying Li [1,2,3,4], Guangxia Wang [1,2,3,4], Jianzhong Guo [1,2,3,4] and Jiayao Wang [1,2,3,4]

[1] College of Geography and Environmental Science, Henan University, Kaifeng 475004, China; zhenghaohua@henu.edu.cn (H.Z.); 10130155@vip.henu.edu.cn (H.L.); wgx@henu.edu.cn (G.W.); 10130151@vip.henu.edu.cn (J.G.); 10130141@vip.henu.edu.cn (J.W.)
[2] Key Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions, Henan University, Ministry of Education, Kaifeng 475004, China
[3] Henan Industrial Technology Academy of Spatio-Temporal Big Data, Henan University, Zhengzhou 450046, China
[4] Henan Technology Innovation Center of Spatio-Temporal Big Data, Henan University, Zhengzhou 450046, China
* Correspondence: jczhang@vip.henu.edu.cn; Tel.: +86-188-1045-0013

Abstract: Selecting road networks in cartographic generalization has consistently posed formidable challenges, driving research toward the application of intelligent models. Despite previous efforts, the accuracy and connectivity preservation in these studies, particularly when dealing with road types of similar sample sizes, still warrant improvement. To address these shortcomings, we introduce a Heterogeneous Graph Attention Network (HAN) for road selection, where the feature masking method is initially utilized to assess the significance of road features. Concentrating on the most relevant features, two meta-paths are introduced within the HAN framework: one for aggregating features of the same road type within the first-order neighborhood, emphasizing local connectivity, and another for extending this aggregation to the second-order neighborhood, capturing a broader spatial context. For a comprehensive evaluation, we use a set of metrics considering both quantitative and qualitative aspects of the road network. On road types with similar sample sizes, the HAN model outperforms other models in both transductive and inductive tasks. Its accuracy (ACC) is higher by 1.62% and 0.67%, and its F1-score is higher by 1.43% and 0.81%, respectively. Additionally, it enhances the overall connectivity of the selected network. In summary, our HAN-based method provides an advanced solution for road network selection, surpassing previous approaches in terms of accuracy and connectivity preservation.

Keywords: cartographic generalization; road network selection; Heterogeneous Graph Attention Network; meta-path

1. Introduction

Cartographic generalization, a fundamental technique in traditional cartography, remains a central focus in the realm of digital cartography, with automatic cartographic generalization standing out as a particularly challenging and innovative research frontier [1,2]. Within the cartographic generalization process, selection serves as the initial and pivotal step. Roads, as essential skeletal elements on maps, not only form a crucial component of maps but also command the attention and interest of both map users and cartographers. Given their significant roles in both social and military contexts, roads have become a focal point of extensive research and practical exploration worldwide, particularly in the realm of automatic road network selection [3–5]. Road network selection methods are mainly divided into unsupervised and supervised methods. The road network selection process is essentially a subjective decision made by cartographers based on objective facts,

and the use of supervised methods to simulate the decision-making process of cartographic experts is the trend of development at this stage. Since graph neural networks can aggregate neighboring road features to embed their own features better, it is an important research direction to use graph neural networks to simulate the decision-making process of road network selection.

In the process of road network selection, the road type data is extremely important, and there is a wealth of interactive information implied between the road types. Previous studies have treated the road type as a feature in the homogeneous graph, ignoring the interactive information between the road types. Dissimilar homogeneous graphs, heterogeneous graphs, or heterogeneous information networks involve a graph data structure encompassing multiple node types or edge types. The exploration of heterogeneous graphs has gained prominence in graph neural network research, as highlighted by studies [6–15].

Heterogeneous graph neural networks serve to depict complex heterogeneous objects and their interactions, offering rich semantic information and an effective modeling tool for graph data mining. Leveraging heterogeneous graph neural networks proves advantageous as they can harness the semantic information of roads. The key challenge in using heterogeneous graphs for road network selection lies in designing a method for road node aggregation that aligns with the characteristics of road selection, essentially, the establishment of a meta-path method. This methodological refinement is critical for optimizing the performance of road network selection models.

It is recognized that roads of the same network type are typically designed to fulfill similar traffic demands during the planning and construction phases, serving comparable transportation purposes. For instance, highways are engineered for high-speed traffic, whereas urban arterial roads are constructed to accommodate urban traffic flow. Moreover, planning and design standards exert a pivotal influence on the correlation between different road types. The norms and standards established at the country, regional, or city level significantly shape the planning and design of roads. Similar planning and design standards often result in the creation of similar road types, thereby increasing their correlation. Given this understanding, roads of the same type exhibit the highest correlation. To capture this correlation, two types of meta-paths have been devised based on the principle of road correlation. This principle facilitates the aggregation of the most pertinent road features. Specifically, one is for aggregating features of the same road type within the first-order neighborhood, emphasizing local connectivity, and another extends this aggregation to the second-order neighborhood, capturing a broader spatial context. The rationale behind designing the second type of meta-path is that a single-layer convolution can effectively aggregate features from adjacent road nodes at a greater distance. Leveraging the flexibility provided by the HAN (Heterogeneous graph attention network) model [10], which allows for the manual setting of meta-paths and employs a du-al-layer attention mechanism for embedding road features, the HAN model is employed to embed road features for these two types of meta-paths. This approach enables a more nuanced representation of road characteristics based on their correlation and enhances the capabilities of the road network selection model.

The remainder of this paper is organized as follows. Section 2 briefly reviews studies related to road network selection. Next, Section 3 proposes an automatic road network selection model based on heterogeneous graphical attention neural networks. Then, Section 4 presents experimental data, results, and a detailed analysis of these results. Finally, Section 5 concludes the paper.

2. Related Work

Road network selection encompasses two key aspects: quantity selection and structure preservation. Quantity selection involves determining how many roads should be chosen, while structure preservation focuses on selecting which roads to include. Common methods for quantity selection include the square root model, correlation analysis, regression analysis, the map-suitable area-load method, and the geometric progression method [16].

The square root model and its variants are generally accepted as effective for quantity selection, shifting attention to the challenge of structure preservation. In terms of structure preservation methods, constraint indicators are commonly considered based on semantics, geometry, and topology information. Structure preservation methods can be categorized into unsupervised and supervised approaches.

Unsupervised selection methods include graph theory method [17–20], block-based methods [21], stroke-based methods [22,23], and road grid density methods [24,25]. These methods primarily focus on the overall morphology of the road network using geometric and topological information. Among them, common improvement methods are based on the stroke-based approach [26]. Some scholars [27] have fused three algorithms (a stroke-based, a grid-based, and a combined stroke-mesh algorithm) with three selection methods and made corresponding improvements. In addition to geometric and topological considerations, road semantic information, such as POI (point of interest), residential areas, and traffic flow data, is also utilized for large-scale urban road network selection based on road functionality [28–31]. Unsupervised methods generally make better use of spatial and semantic information but require a deep understanding of the interactions between various road information and the formulation of selection rules. However, these methods are less automated because of their high subjectivity in determining constraint indicators and their weights.

The supervised selection methods of road networks have become a focal point of current discussions. With the rapid evolution of artificial intelligence, various techniques, including kernel machine learning [32], BP (backpropagation) neural networks [33], Radial Basis Function [34], decision-tree-based (DT) models [35], and ontology knowledge reasoning [36], have been progressively applied to automate road network selection. Moreover, it has been demonstrated that road networks selected using machine learning models are extremely similar to the atlas maps [37]. Despite these advancements, most supervised selection methods overlook the topological information inherent in road networks. Transforming the road network into a graph structure and employing graph theory in the automatic selection process has gained attention [38–41].

Leveraging the substantial achievements of deep convolutional neural networks in image recognition, Kipf and Welling [42] extended these networks to graph data, resulting in the development of Graph Convolutional Networks (GCN). While GCN effectively utilizes the topology information, its limited generalization ability, relying on the Laplacian operator, led scholars to introduce attention mechanisms into graph convolution, giving rise to Graph Attention Networks (GAT) [43].

The growing demand for graph convolution operations on large-scale graphs prompted the introduction of Graph Sampling Aggregation Networks (GraphSAGE) [44]. Dissimilar methods aggregate all neighboring nodes, and GraphSAGE samples a subset of road neighboring nodes for aggregation, making it suitable for large-scale graphs. Scholarly studies have found that graph convolutional neural networks can learn spatial relationships [45]. Zhang et al. applied graph convolutional neural networks to automatic road network selection, comparing the performance of three homogeneous graph convolutional networks: GCN, GAT, and GraphSAGE [46]. Addressing the issue of gradient vanishing in deep graph neural networks, Zheng et al. incorporated architectures such as JK-Nets, ResNet, and DenseNet into road network selection, resulting in significant performance improvements [47]. Additionally, Ma et al., Zhu et al., and Guo et al. utilized road stroke data to construct spectral domain graph convolution operators for road network selection [48–50]. Some scholars have also applied graph neural networks to the selection of rivers and achieved good results [51,52].

Despite the use of graph neural networks (GNNs) in previous studies for automatic road network selection, several shortcomings persist:

- Not exploring the importance of each feature of graph neural networks in the road network selection task;
- Performance Gaps in Intermediate-Grade Roads: Homogeneous graph selection algorithms exhibit poor performance, particularly concerning intermediate-grade roads. Additionally, there is a need to enhance the overall connectivity of the selected road network;
- Lack of Comparison in Transductive and Inductive Tasks: Previous studies have not adequately compared the selection performance of the model in both transductive and inductive tasks. Transductive tasks involve training a model using road data with a small number of known labels to infer the majority of the remaining labels. Inductive tasks, on the other hand, entail training a model using road data with numerous known labels to predict labels for nodes on a new road dataset. Such comparisons are essential for a comprehensive evaluation of the selection model, especially when considering road au-to-selection tasks in different spatial domains.

Addressing these shortcomings will contribute to the development of more robust and versatile graph neural network models for automatic road network selection.

3. Methods

We recognize the limitations of prior intelligent models in the road network selection task; our proposed approach addresses these shortcomings through a multi-step process. Initially, the dual graph is generated following data processing. Within the framework of a graph neural network, the significance of road features is evaluated using the feature masking method. Subsequently, the HAN model is introduced to refine the road network selection results. This is achieved by strategically designing two types of meta-paths, which capitalize on the road correlation principle. Finally, to validate the effectiveness of our proposed method, comprehensive comparisons are conducted against other classical intelligent models. The assessment encompasses both transductive and inductive tasks in the context of road selection. This rigorous evaluation serves to demonstrate the superiority and versatility of our proposed method in improving the precision and adaptability of road network selection processes. The detailed automatic road network selection process is illustrated in Figure 1.

3.1. Measurement of Road Feature Importance Based on Feature Masking Method

During the construction of graph data for road networks, the subjective nature of road feature selection introduces potential issues such as data redundancy, impacting model accuracy. Hence, it is imperative to delve into the significance of different road features in the road selection model. These features primarily stem from three dimensions: geometric, semantic, and topological.

Semantic features encompass road types, while geometric features comprise road length, the number of road vertices, road aspect ratio, mesh density, curvature ratio, and the coordinates of start and end points (X, Y). Topological features include degree, degree centrality, eigenvector centrality, betweenness centrality, and closeness centrality. For detailed information, please consult Table 1. Despite the ability of graph neural networks to capture road topology, the consideration of global topological features is currently absent. Consequently, this study also explores the significance of road topological features.

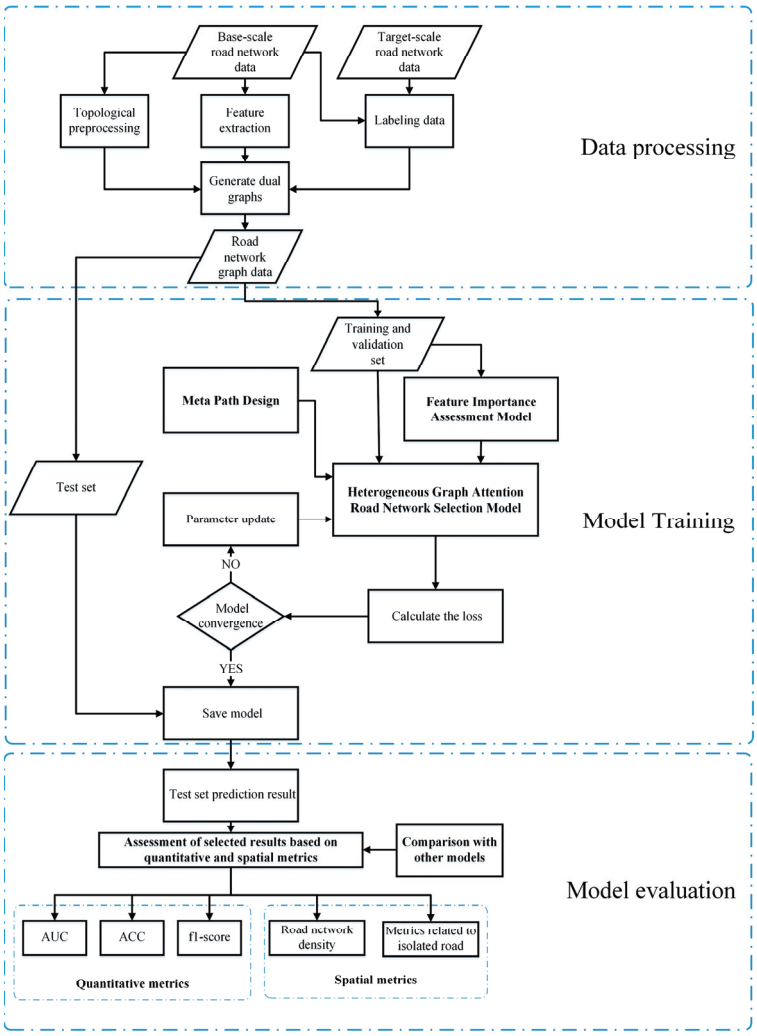


Figure 1. Flowchart of road network selection.

Table 1. Road network features.

Feature Types	Feature Indicators	Detailed Explanation
Semantic feature	road type	Road type is a system that classifies roads according to characteristics such as traffic flow, scale, and function.
Geometric features	road length	The length of roads in projected coordinates.
	number of road vertices	The number of vertices in each road polyline.
	road aspect ratio	The ratio of the length of the road's horizontal coordinates to its vertical coordinates.
	mesh density	The maximum ratio of the perimeter to the area of the left and right polygons associated with each road (if there are no left and right polygons, the value is set to 0).
	curvature ratio	The ratio of the road length to the straight-line length between the start and end coordinates of the road.
	start and end points (X, Y)	Start and end point coordinates (four values in total).

Table 1. *Cont.*

Feature Types	Feature Indicators	Detailed Explanation
Topological features	degree	The degree of each road is equal to the number of intersections it has with other roads.
	degree centrality	The degree of each road is divided by the total number of roads minus one.
	eigenvector centrality	The eigenvector corresponding to the largest eigenvalue of the adjacency matrix represents the centrality of the eigenvector for each node.
	betweenness centrality	The ratio of the number of times the shortest paths between all other pairs of nodes pass through a particular node to the total number of shortest paths in a graph.
	closeness centrality	The total number of nodes minus one divided by the total number of shortest paths from that node to other nodes.

We used a relatively simple feature masking method to assess the importance of road features. Initially, all features slated for evaluation are fed into a two-layer Graph Attention Network (GAT) [43] model for training. The final trained model is determined by selecting the optimal Area Under the Curve (AUC) on the validation set. Subsequently, each feature under scrutiny is individually set to 0, and the ensuing change in the AUC metric on the validation set is observed. The evaluation of road feature importance is contingent upon the observed AUC variation. A higher decrease in the AUC value indicates greater importance for the corresponding road feature.

3.2. Construction of HAN Model for Road Network Selection

3.2.1. Meta-Path Design Method Based on Road Correlation

Before embarking on meta-path design, it is crucial to establish the methodology for transforming the road network into graph data. In our proposed method, the road network type is designated as the node type in the graph data structure, leveraging the advantages of graph neural networks in embedding node features and aggregating features of neighboring nodes associated with them. In this context, intersections are treated as connecting edges, while roads are construed as nodes within the dual graph. Consequently, the original road network selection transforms into a question of node classification.

Figure 2 illustrates the transformation of a road network into a dual graph, using the U.S. 1M scale road network as an illustrative example. The four road types carry specific meanings:

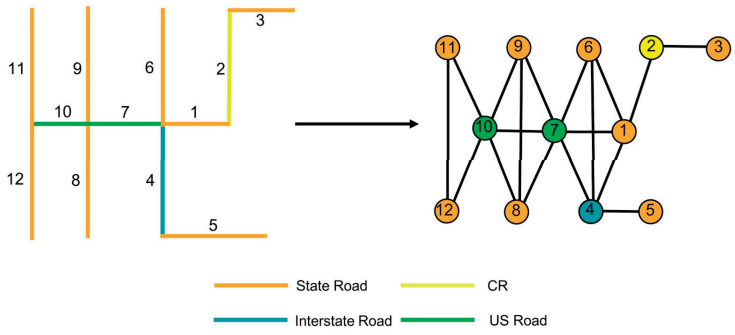


Figure 2. Constructing dual graph data based on road network intersections.

These distinctions lay the foundation for a nuanced representation of the road network, enabling effective node classification and guiding subsequent meta-path design.

- State Road: Highways constructed and maintained by state governments, facilitating connections between cities and regions within a state.
- US Road: Before the construction of the Highway System, US Roads served as the primary highway network. Nowadays, they continue to play a crucial role as major transportation corridors within states and between regions.
- Interstate Road: The primary objective of an Interstate Road is to enhance the safety and efficiency of automotive travel. Generally, Interstate Roads permit the fastest speeds compared with any other roadways in the vicinity.
- CR: County roads constructed and maintained by local governments serve as vital conduits linking cities and regions within a county.

In the HAN model, the construction of the adjacency matrix deviates from the homogeneous graph convolution algorithm. Specific meta-paths need to be devised for each category of road nodes, enabling the generation of distinct adjacency matrices to capture various semantics. A meta-path, an abstract path pattern delineating node relationships in intricate networks, aids computer models in comprehending network structures. We explain the role of meta-path in terms of road type as shown in Figure 3 below:

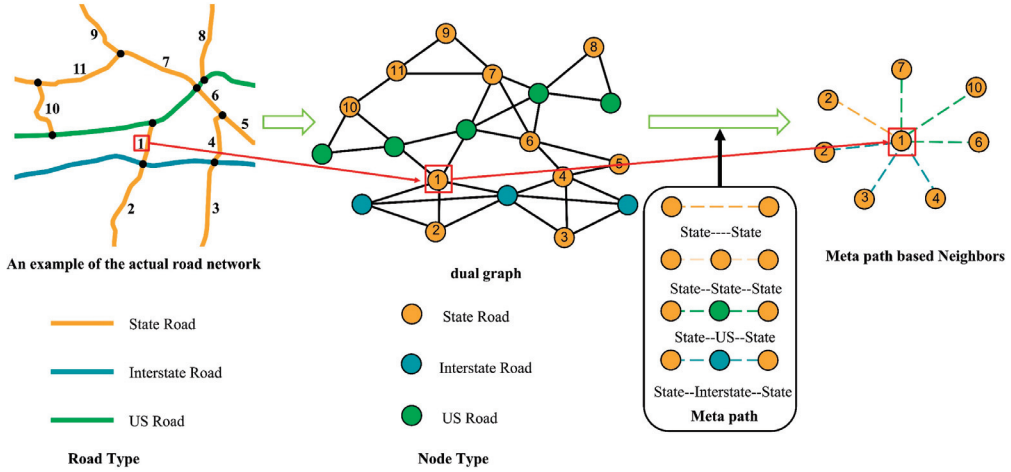


Figure 3. An illustrative example of a meta-path.

Since road segments of the same type often exhibit high correlation, the second-order neighbors belonging to the same road type as a given road segment positively influence its road node embedding. In this study, we utilize first-order neighbor roads as bridges to generate distinct meta-paths. Importantly, meta-paths formed by different types of intermediate bridge roads convey distinct semantic information. For instance, the meta-path 'State-Interstate-State' signifies traveling from a State road to an Interstate road and then back to a State road, thus connecting two second-order neighbors of the same road type. Meta-paths enable a deeper comprehension of information flow within intricate networks, thereby enhancing model performance. By searching for semantically connected neighbor nodes, meta-paths facilitate the aggregation of neighboring node features, leading to more effective embeddings of road segment characteristics.

Graph neural networks outperform traditional neural networks in node classification by aggregating neighboring node features to create more robust node embeddings. The core principle involves aggregating node features closely linked to the target node, incorporating the features of neighboring nodes with strong connections. Following this methodology, meta-paths for road networks can be crafted to aggregate features of closely related road nodes.

Given that roads of the same type generally adhere to similar design standards, functions, and purposes, they are inherently more closely related. Consequently, the connectivity and coherence of roads are typically considered during the planning and construction processes.

Two distinct types of meta-paths were formulated for each road type. The first type consolidates node features of the same road type within the first-order neighbors, while the second type encompasses node features of the same road type within the second-order neighbors. In the case of N road types, N + 1 meta-paths are established. In instances where there is no road type attribute, the road network can be hierarchically divided using the back-propagation method based on semantic levels [53]. Subsequently, the corresponding adjacency matrix can be generated following the design principles of meta-paths outlined in the previous section.

Taking the example of roads labeled as 'State' in Figure 2, five meta-paths are identified: ① State-State, ② State-State-State, ③ State-US-State, ④ State-Interstate-State, and ⑤ State-CR-State. In summary, two primary categories of meta-paths are developed. The first type aggregates features of the same road type within the first-order neighbors, exemplified by ①. The second type aggregates features of the same road type within the second-order neighbors, illustrated by ②, ③, ④, and ⑤. Consequently, the four road types yield five distinct adjacency matrices. Using the node labeled '1' in Figure 2 as an illustration, the surrounding neighboring nodes aggregated under each meta-path are detailed in Table 2.

Table 2. The indices of the neighboring nodes that need to be aggregated under each meta-path.

Meta-Path	Indices of Neighboring Nodes to Be Aggregated
State-State	6
State-State-State	none
State-US-State	6, 8, 9
State-Interstate-State	5, 6
State-CR-State	3

3.2.2. Heterogeneous Graph Attention Network Embedding Road Features

The Heterogeneous Graph Attention Network is a deep learning model designed to handle heterogeneous graph data characterized by diverse node and edge types. In the context of transforming road networks into graph data, the heterogeneity is manifested in the varied types of roads. Dissimilar to traditional road selection models that typically focus on homogeneous graphs with nodes and edges of the same type, the HAN model autonomously configures meta-paths and employs both node-level and semantic-level attention mechanisms.

Node-level attention refers to the mechanism within a HAN that assigns different weights to neighboring nodes based on their importance to the target node. Given a specific meta-path, a node may have multiple neighbors. Node-level attention aims to distinguish the informative neighbors from the less informative ones by measuring their importance and assigning them corresponding attention weights. In the context of road network embedding, node-level attention determines the aggregation weights when combining the features of neighboring roads under a particular meta-path for each road.

Semantic-level attention, on the other hand, focuses on measuring the importance of different meta-paths. Since different meta-paths can carry different semantic information, it is crucial to determine which meta-paths are more informative for a specific task. Semantic-level attention assigns weights to different meta-paths based on their relevance and importance. In the road embedding scenario, semantic-level attention determines the weight coefficients used in the weighted summation of road embeddings derived from different meta-paths.

To further explain node-level attention and semantic-level attention, Figure 4 shows how the node-level and semantic-level attention mechanisms are aggregated after the road network is transformed into a dual graph.

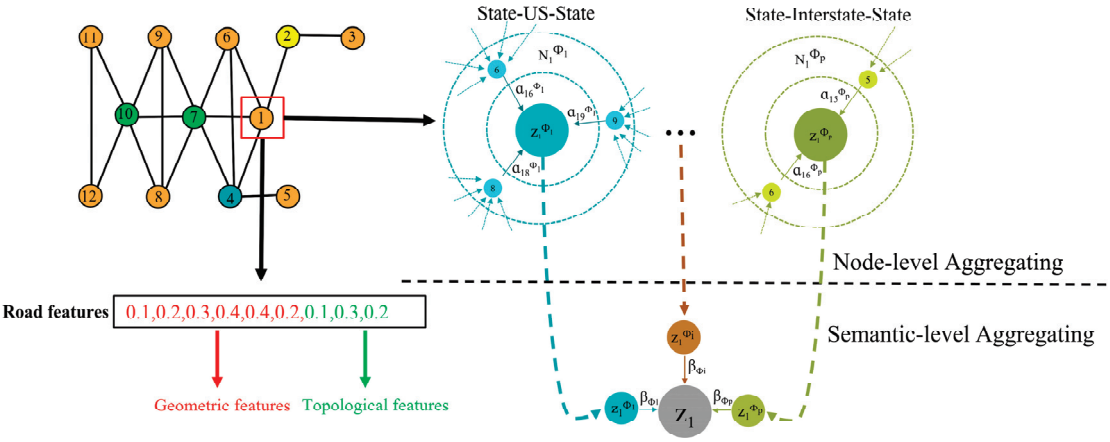


Figure 4. Explanation of aggregating process at both the node-level and the semantic-level.

The HAN model is particularly adept at aggregating road neighboring features under two types of meta-paths. Semantic-level attention in road networks is influenced by road types, leading to varying attention coefficients as road features are embedded and aggregated along different meta-paths. Simultaneously, the node-level attention takes into account the attention weights of different road features from neighboring roads when aggregating features within the same meta-path. The node-level attention [10] coefficient is calculated as:

$$\alpha_{ij}^{\Phi} = softmax_j\left(e_{ij}^{\Phi}\right) = \frac{\exp\left(\sigma\left(\mathbf{a}_{\Phi}^{\mathrm{T}} \cdot \left[\mathbf{h}_i' \parallel \mathbf{h}_j'\right]\right)\right)}{\sum_{k \in \mathcal{N}_i^{\Phi}} \exp\left(\sigma\left(\mathbf{a}_{\Phi}^{\mathrm{T}} \cdot \left[\mathbf{h}_i' \parallel \mathbf{h}_k'\right]\right)\right)} \tag{1}$$

where the notation $\parallel$ represents the concatenation operation, $\mathbf{h}_i'$ and $\mathbf{h}_j'$ denote the features of its road and the features of the neighboring roads under this meta-path, respectively. Additionally, $\mathbf{a}_{\Phi}$ represents the shared attention weights for the meta-path Φ. The notation σ denotes the activation function, and the attention coefficients are normalized through a SoftMax layer.

The semantic-level attention [10] coefficients are calculated as follows:

$$w_{\Phi_p} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{q}^{\mathrm{T}} \cdot \tanh\left(\boldsymbol{W} \cdot \boldsymbol{z}_i^{\Phi_p} + \boldsymbol{b}\right) \tag{2}$$

$$\beta_{\Phi_p} = \frac{\exp\left(w_{\Phi_p}\right)}{\sum_{p=1}^{P} \exp\left(w_{\Phi_p}\right)} \tag{3}$$

$\mathcal{V}$ represents the set of all road nodes, $\mathbf{q}$ is the semantic level attention vector, $\boldsymbol{W}$ is the weight matrix, and $\boldsymbol{b}$ is the bias vector, which are the three trainable parameters shared for all meta-paths. $\boldsymbol{z}$ denotes the node level embedding vector of the ith node under that meta-path. As shown in Equation (3), w_{Φ_p}, which is obtained under all meta-paths, is normalized to obtain the semantic level attention coefficients finally.

In the proposed model, the initial step involves converting the road network into a dual graph. Subsequently, the spatial geometry features, semantic features, and topological features of the roads are amalgamated to construct a feature matrix. In this matrix, each row corresponds to a road, and each column represents a numericalized feature derived from the three types of features. Node-level attention coefficients play a pivotal role in aggregating neighboring road network features for each meta-path, facilitating the generation of node-level embeddings for the road network.

Following this, semantic-level attention coefficients are employed to aggregate the node-level embedding features obtained from each meta-path. This process culminates in

the creation of semantic-level embeddings for the road network. It is important to note that the proposed model undergoes training using backpropagation and gradient descent, allowing for iterative refinement of the model parameters to enhance its performance.

3.3. The Framework of the HAN Model

The framework of the HAN model is depicted in Figure 5. The model is composed of an input layer, a block of graph convolution layers, a fully connected layer, a prediction probability result layer, and a prediction result layer.

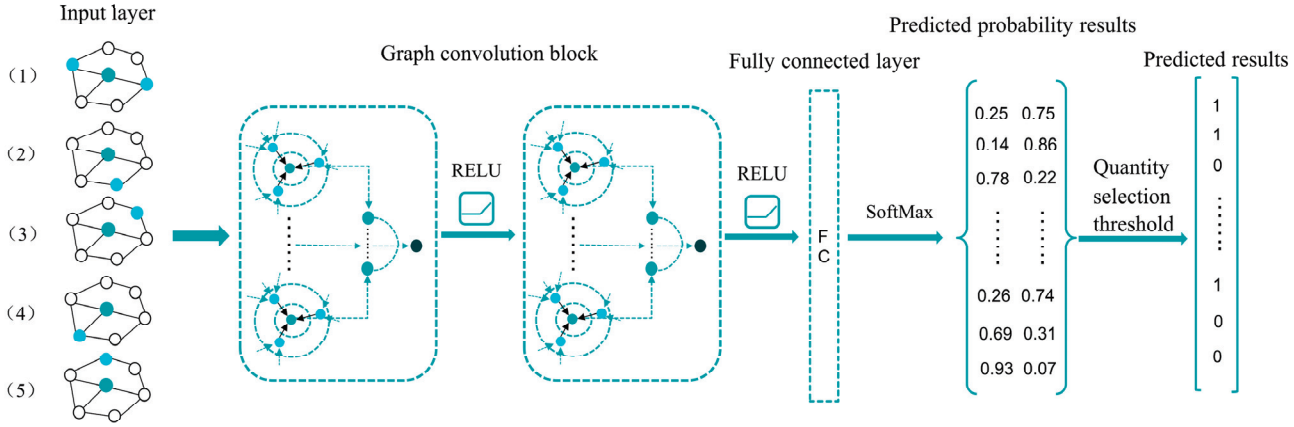


Figure 5. The framework of the HAN road network automatic selection model((1)-(5) Serial numbers represent 5 meta paths).

Taking the U.S. 1 M scale road network as an example, the input layer represents five adjacency node graphs denoted by (1)–(5), each requiring aggregation under five distinct meta-paths. The HAN model's graph convolution module primarily involves aggregating node-level features using computed node-level attention coefficients and semantic-level features using semantic-level attention coefficients. The graph convolution block comprises two convolution layers, utilizing the ReLU function as a non-linear activation function and applying Dropout for regularization.

The fully connected layer compresses the feature matrix into a prediction result matrix of size CR*RR. CR refers to the row number of the matrix, indicating a specific road for prediction purposes, while RR represents the number of columns in the matrix, which corresponds to the respective probabilities of selecting or not selecting a particular road. The prediction probability result layer indicates the likelihood of selecting or not selecting each road segment. The prediction result layer determines the selection status of a road segment based on a selection threshold, where 1 indicates selection and 0 indicates non-selection.

The model is trained using the backpropagation gradient descent algorithm, with cross-entropy loss serving as the training objective and the Adam optimizer employed for the training process.

3.4. Evaluation Metrics for the HAN Model

The assessment of road network selection quality is primarily centered on validating the preservation of spatial distribution characteristics, ensuring an appropriate density, and scrutinizing potential disruptions to road connectivity, among other considerations. Spatial distribution characteristics encapsulate the overall features of the road network, providing a comprehensive overview of its structure and layout. In contrast, density characteristics emphasize local attributes, focusing on the density and compactness of roadways within specific areas. Preserving these characteristics necessitates subjective qualitative evaluation, enabling a thorough understanding of the road network's layout and functionality.

The evaluation of road network selection results in this paper predominantly encompasses quantitative metrics and an assessment of spatial distribution. The AUC, ACC, and

f1-score indicators serve as the primary metrics for evaluating the quantitative aspects of road network selection results. The AUC metric identifies the optimal model from the validation set, while the ACC and f1-score metrics gauge the model's proficiency in automatically selecting road networks. Furthermore, the evaluation includes road network density, as well as the number of isolated roads and their total length, to appraise the spatial distribution quality of the selection result.

3.4.1. Evaluation Metrics for Quantity Assessment of the HAN Model

AUC, ACC, and f1-score stand as widely employed evaluation metrics for binary classification models. ACC signifies the proportion of correctly predicted labels to the total number of samples. The ACC value serves as an indicator of the model's quality. Its calculation formula is expressed as follows [47]:

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

where True Positive (TP) denotes the instances where both predicted and true values are positive, indicating true positive cases and False Positive (FP) represents instances where the predicted value is positive while the true value is negative, indicating false positive cases. False Negative (FN) signifies instances where the predicted value is negative while the true value is positive, indicating false negative cases. True Negative (TN) denotes instances where both predicted and true values are negative, indicating true negative cases.

The calculation principle of AUC involves determining the area under the Receiver Operating Characteristic (ROC) curve. This curve is formed by adjusting the probability threshold from 0 to 1, with the False Positive Rate (FPR) on the X-axis and the True Positive Rate (TPR) on the Y-axis. The calculation formulas are as follows [47]:

$$FPR = \frac{FP}{FP + TN} \quad TPR = \frac{TP}{TP + FN} \tag{5}$$

Furthermore, given the imbalanced distribution of positive and negative labels in road network data, AUC is employed as the evaluation metric for the road selection model on the validation set.

The f1-score, as the harmonic mean of recall and precision, provides an assessment of the model's performance on positive samples. In this study, the f1-score is utilized to evaluate the model's performance, specifically on positive samples. The formula for calculating the f1-score is expressed as [47]:

$$f1_score = \frac{2Precision * Recall}{Precision * Recall} \tag{6}$$

The calculation formulas for recall and precision are defined as follows:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \tag{7}$$

3.4.2. Road Network Density

Road network density is characterized as the ratio of the total length of roads within a specific area to the total area of that region, providing a measure of road density within the designated area. The calculation for road network density is described by [47] as:

$$\rho = \frac{L_{sum}}{S} \tag{8}$$

where L_{sum} represents the sum of the lengths of all roads within the specific area, S represents the area of a specific region. In this study, the road network density is selected as an evaluation metric for the model to assess its performance. The primary objective

is to compare the road network density chosen by experts with that of the automatically selected road model.

3.4.3. Metrics Related to Isolated Road

To assess the connectivity of the chosen road network, both the total number and length of isolated roads are taken into account. The following method is employed to derive these metrics: Initially, the road network selected by the automatic selection model is transformed into graph data, focusing on road intersections. Subsequently, the largest connected subgraph within these graph data is extracted. Finally, roads within this largest connected subgraph lacking road intersections are identified as isolated roads. UtilizingArcGIS 10.2, the total length and count of isolated roads are computed. Lower values for the total length and number of isolated roads indicate enhanced connectivity of the selected roads.

4. Experimental Process and Results

4.1. Experimental Data and Data Preprocessing

Experimental data in this study comprise road network data of the United States, sourced from the USGS and encompassing both 1 M and 2 M scale datasets. The experimental area chosen for analysis consists of nine eastern states of the United States, characterized by intricate road networks. These states include Illinois, Indiana, Kentucky, Ohio, West Virginia, Virginia, Pennsylvania, North Carolina, and parts of Maryland, as illustrated in Figure 6. The road network at scales of 1 M and 2 M within the study area is visualized in Figure 7.

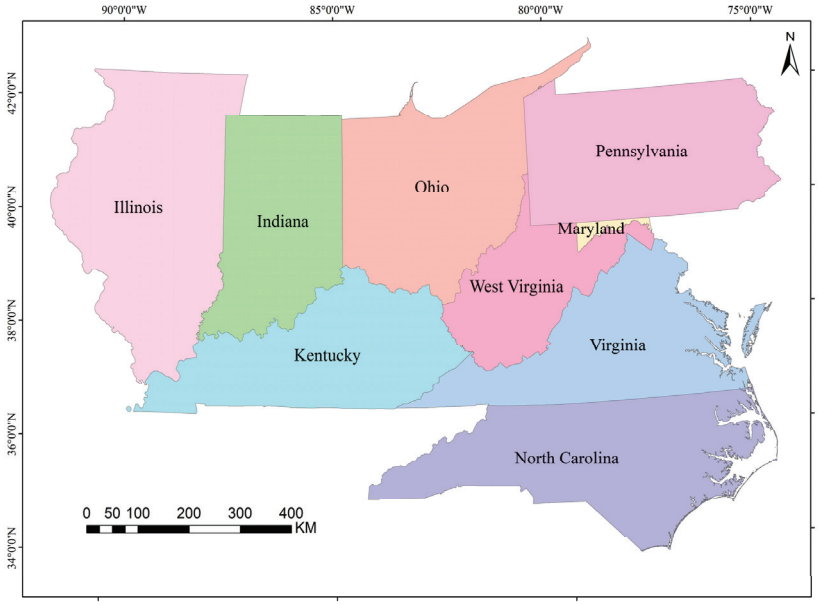


Figure 6. Study area.

To investigate the significance of road features, this study employed data from the first seven states and portions of Maryland. Employing an 8:2 ratio, positive and negative samples of various road types were randomly stratified and divided into training and validation sets. The count of positive and negative samples for each road type is presented in Table 3, while the spatial distribution of the two datasets is visualized in Figure 8.

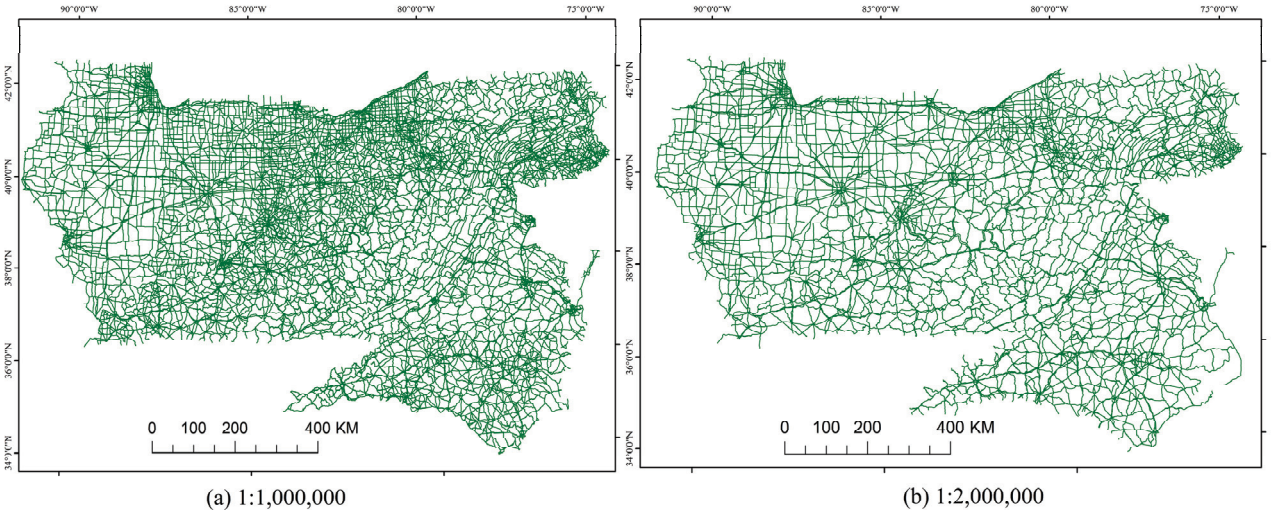


Figure 7. Two scales of road network in the study area: (**a**) 1:1,000,000 Road network data; (**b**) 1:2,000,000 Road network data.

Table 3. Quantitative statistics of road data for feature importance exploration.

Road Types	Positive Training Samples	Negative Training Samples	Positive Validation Samples	Negative Validation Samples	Total
State	4840	4151	1221	1042	11,254
US	4205	266	1030	63	5564
Inter	1861	64	471	15	2411
CR	14	18	7	4	43

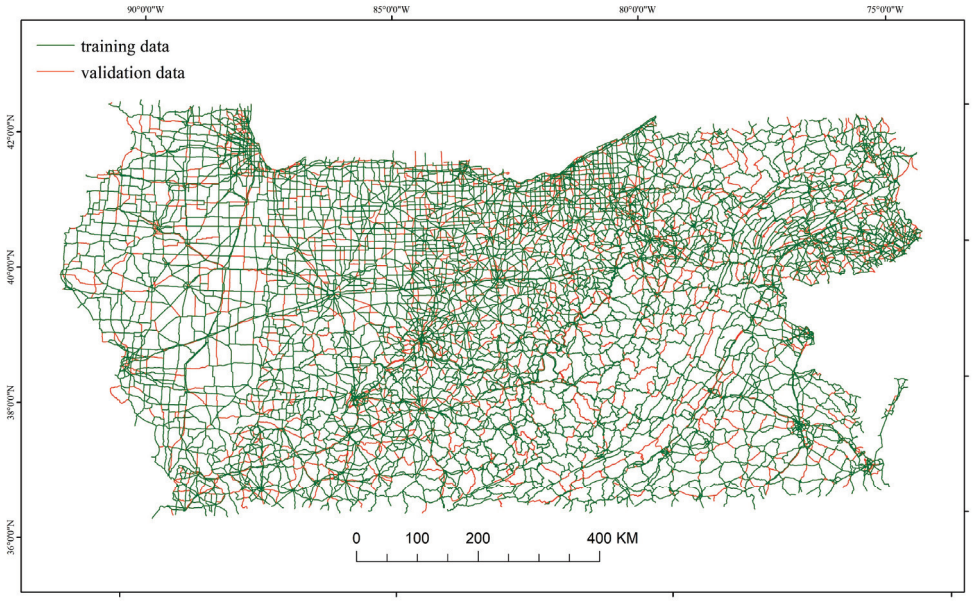


Figure 8. Spatial distribution of road data for feature importance exploration.

For the transductive road selection task, the division of training, validation, and testing sets was performed through stratified random sampling of positive and negative samples in a ratio of 2:1:7. The spatial distribution of these three datasets is illustrated in Figure 9, and the corresponding quantitative statistics are presented in Table 4.

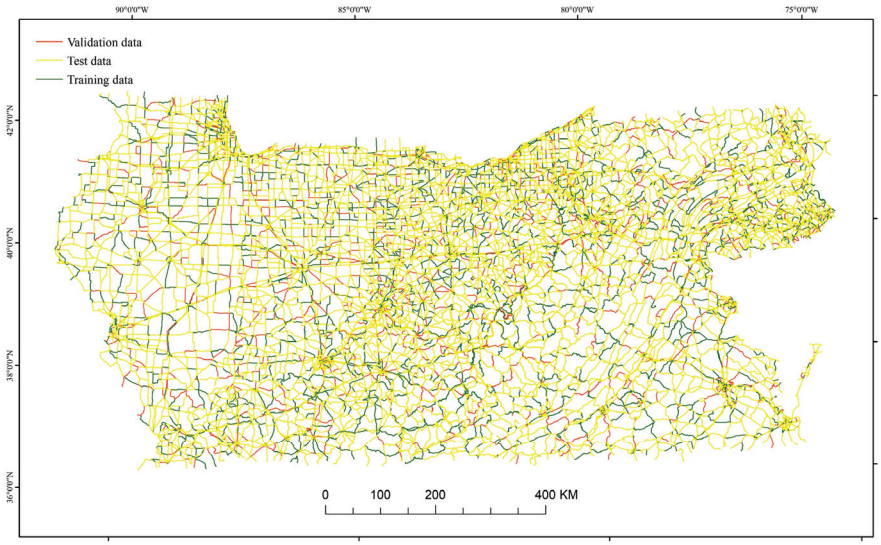


Figure 9. Spatial Distribution of road data for the transductive road network selection model.

Table 4. Quantitative statistics of road data for the transductive road network selection model.

Road Types	Positive Training Samples	Negative Training Samples	Positive Validation Samples	Negative Validation Samples	Positive Testing Samples	Negative Testing Samples	Total
State	1205	1037	607	518	4249	3638	11,254
US	1061	64	517	34	3657	231	5564
Inter	462	20	241	9	1629	50	2411
CR	1	3	1	2	19	17	43

For the inductive road network selection model, the training and validation sets consist of positive and negative samples divided using stratified random sampling in an 8:2 ratio. The testing set is exclusively from North Carolina. The spatial distribution of the three datasets is depicted in Figure 10, and the quantity distribution is outlined in Table 5.

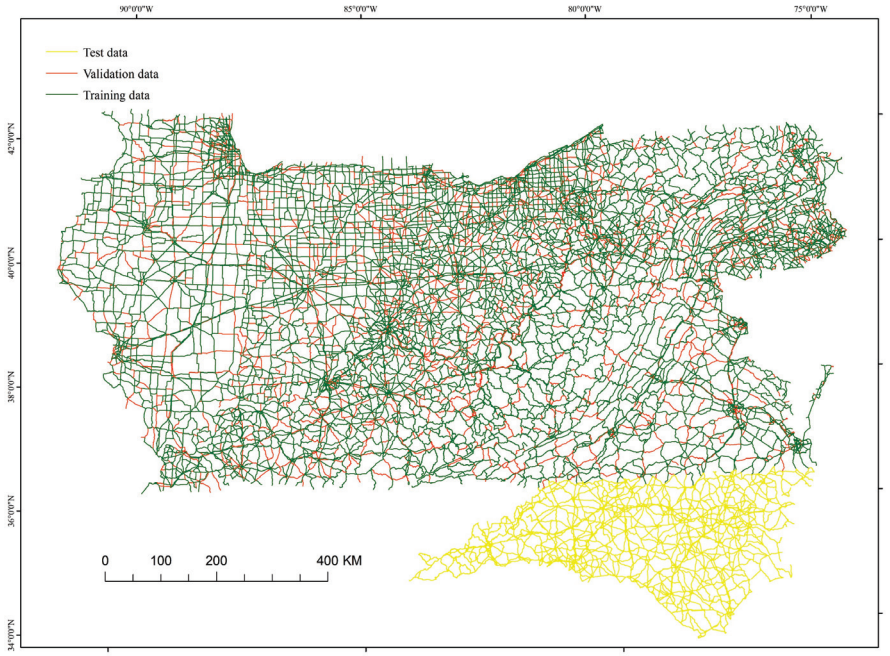


Figure 10. Spatial distribution of road data for the inductive road network selection model.

Table 5. Quantitative statistics of road data for the inductive road network selection model.

Road Types	Positive Training Samples	Negative Training Samples	Positive Validation Samples	Negative Validation Samples	Positive Testing Samples	Negative Testing Samples	Total
State	4869	4162	1192	1031	495	702	12,451
US	4165	255	1070	74	1095	133	6792
Inter	1864	63	468	16	268	46	2725
CR	21	18	0	4	0	1	44

Data preprocessing encompasses the removal of isolated road segments and the execution of topological construction. For 1 M road types, missing road types are imputed based on the road types connected to them in 2 M road data. Label construction for these training data primarily involves generating buffers at two different scales and then assigning labels based on the coverage area ratio of these buffers. Road network data slated for selection are assigned a label attribute of 1, indicating positive samples, while the remaining road network data receive a label attribute of 0, denoting negative samples. Following manual verification and adjustment of data labels, intersecting lines are segmented. Subsequently, the geometric and topological features of each road segment are computed, and ultimately, adjacency and feature matrices are constructed.

4.2. Results of Road Feature Importance Measurement

To assess the significance of road features in the divided dataset, a two-layer Graph Attention Network (GAT) road network auto-extraction model is employed, aligning with the framework of the HAN model. In the construction of the feature matrix for the GAT model, road types are encoded using one-hot encoding. Initially, all features are fed into the GAT model, featuring a hidden layer with 64 neurons, 4 attention heads, and a dropout rate of 0.1. The training steps and weight decay coefficient are manually adjusted. The model undergoes training for 1000 epochs, and the model exhibiting the highest AUC on the validation set is chosen as the final benchmark model, achieving an optimal AUC of 0.8413. Subsequently, the evaluated features are set to 0 in the feature matrix, and the trained model is employed to observe the change in AUC, thus gauging the contribution of each feature to the model. The experimental results are presented in Table 6.

Table 6. The statistical results of AUC on the validation set are based on the feature masking method.

Road Types	Road Length	Mesh Density	Number of Road Vertices	Road Aspect Ratio	Curvature Ratio
0.6108	0.7902	0.8414	0.8417	0.8413	0.8131
Betweenness Centrality	**Eigenvector Centrality**	**Closeness Centrality**	**Start and End Points X, Y**	**Degree**	**Degree Centrality**
0.8399	0.8414	0.7767	0.6449	0.8056	0.7850

The AUC changes in road features that positively impact the GAT model are consolidated, and the proportion of each feature in the total sum is calculated. The results are visually represented in Figure 11.

The results indicate that road type holds the highest importance, followed by start and end points (X, Y), closeness centrality, degree centrality, road length, degree, curvature ratio, and betweenness centrality. Table 7 showcases the selection outcomes of the GAT model on the validation dataset, utilizing the partition threshold established by the total count selected by experts.

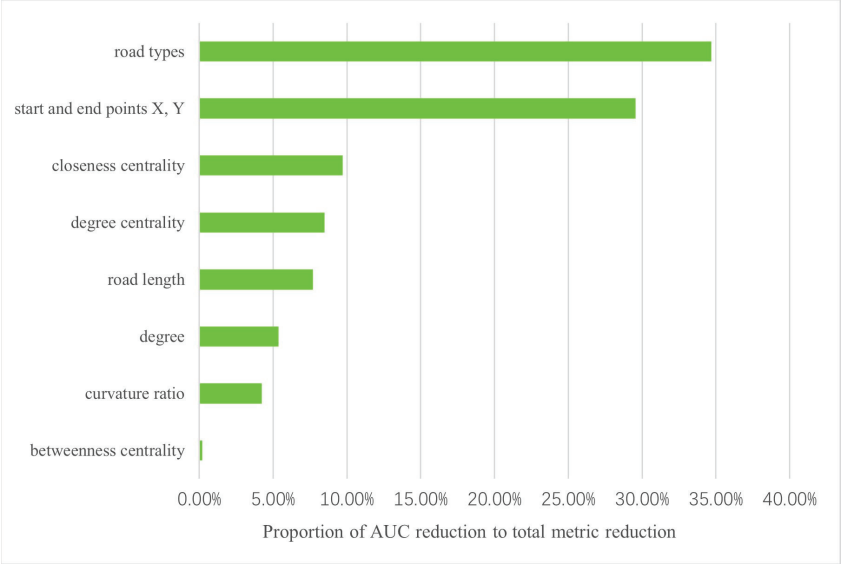


Figure 11. Ranking of road features with positive impact importance.

Table 7. Accuracy of various road types in the GAT validation set.

Road Type	State Road	US Road	Interstate Road	CR
ACC	0.6681	0.9424	0.9691	0.7273

The GAT model exhibits insufficient performance for the State and CR road types, with limited discussion on the CR road type due to its constrained data volume. Particularly for the State Road type, characterized by a relatively balanced distribution of positive and negative samples, the GAT model's performance falls short of expectations. Consequently, there arises a need to develop dedicated automatic selection models tailored to specific road types within the road network. In response, the HAN model is introduced for the automatic selection of roads, enabling classification-based selection. The subsequent automatic road network selection model selects start and end points X, Y, closeness centrality, degree centrality, road length, degree, and curvature ratio as features for model training.

4.3. Analysis of the Results of the Transductive and Inductive Road Network Selection Task

To evaluate the selection performance of the HAN model, it is compared with three other models: the Multi-layer Perceptron (MLP), the homogeneous graph neural network GAT, and the Fast Graph Transformer Networks (FastGTN) [14]. The MLP model serves as a baseline without graph convolution, enabling us to assess the impact of graph convolutions on performance. The GAT model incorporates graph convolution but without meta-path aggregation, helping us understand the effect of this aggregation on road network selection. Last, the FastGTN model generates all meta-paths of arbitrary length, allowing for a comparison to derive performance differences between arbitrarily generated meta-paths and our designed meta-paths.

4.3.1. Analysis of the Road Selection Results in the Transductive Task

The transductive task involves training a model on these same graph data with a small number of known labels to infer the majority of the remaining labels. This task requires the model to generalize beyond the training samples, especially when dealing with limited labeled data. The application of this task to road selection occurs when only a few road segments are known, and the objective is to predict the selection of the remaining road segments within the same road network.

The parameters for the various models are configured as follows: The MLP model features 512 hidden layers, while both the HAN and GAT models have hidden layers with

64 neurons, 4 attention heads, and a dropout rate of 0.1. Conversely, the FastGTN model is characterized by hidden layer neurons set to 128 with 4 channels. In the FastGTN model, the initial matrix considers the adjacency of all road types, resulting in a total of N^2 initial matrices, where each road type is adjacent to N other road types.

The weight decay coefficients and training step lengths are manually adjusted, and the training process is conducted for 1000 epochs. The final model is selected based on the optimal AUC on the validation set, which is suitable for imbalanced binary classification problems, and performance metrics are compared on the test set. Since the model output represents selection probabilities when aggregated from 1 M to 2 M, the threshold for the selection quantity is determined based on road data of two different scales.

The HAN model selects roads based on road types individually, and all models in this study use expert-selected quantities for various road types to establish selection thresholds. Given the limited quantity of CR-type roads, the HAN model cannot be adequately trained for this specific road type. Consequently, individual comparisons for CR roads are not conducted, and the overall selection results for CR roads in the HAN model are computed based on the GAT model's selection results. To standardize variables, CR-type roads in other models are also based on the GAT model's selection results.

In these experimental road network data, the number of positive and negative samples for state roads is relatively balanced. Table 8 illustrates that the HAN model's selection for State-type roads surpasses other models in terms of ACC and f1-score. Specifically, ACC and f1-score are 1.62% and 1.43% higher, respectively, compared with the other models. Overall, for the selection by the HAN model, ACC and f1-score are 1.02% and 1.01% higher, respectively, than other models. In terms of the quantity and total length of isolated roads, the HAN model records 158 fewer roads and 1414.44 km less than other optimal models.

Table 8. The statistical results of road selection models in the transductive task.

Evaluation Metrics	Model	All	State	US	Interstate
ACC	HAN	0.7535	0.6416	0.9051	0.9363
	GAT	0.7385	0.6163	0.8976	0.9517
	FastGTN	0.7433	0.6254	0.8968	0.9488
	MLP	0.7316	0.6120	0.8876	0.9398
f1 score for positive samples	HAN	0.8257	0.6664	0.9495	0.9671
	GAT	0.8152	0.6440	0.9455	0.9751
	FastGTN	0.8156	0.6521	0.9451	0.9735
	MLP	0.8104	0.6395	0.9402	0.9710
Isolated roads (number | total length)	HAN	40 | 343.01 km	26 | 278.57 km	4 | 9.80 km	10 | 54.64 km
	GAT	198 | 1757.45 km	61 | 712.64 km	130 | 998.95 km	7 | 45.87 km
	FastGTN	281 | 1342.01 km	243 | 990.96 km	38 | 351.06 km	0 | 0 km
	MLP	245 | 942.37 km	104 | 362.05 km	80 | 338.94 km	61 | 241.37 km
Road network density (km/km^2)	Expert Selection	0.13513	0.06675	0.04855	0.01961
	HAN	0.14212	0.07309	0.04912	0.01938
	GAT	0.14591	0.07721	0.04845	0.01973
	FastGTN	0.13784	0.0682	0.04934	0.01971
	MLP	0.13661	0.0692	0.04746	0.01939

The comparison of the selection results of the four models with the expert selection results is depicted in Figure 12. The four models tend to under-select the middle of the road network, although the HAN model and the GAT model show improvement. Additionally, the HAN model exhibits the lowest number of isolated roads among the four models. The main reason why most of the isolated roads appear at the edges of the road network is because the cropping operation of the road network destroys the original topological features of the roads at the margins, so it produces many isolated roads appearing at the margins of the road network.

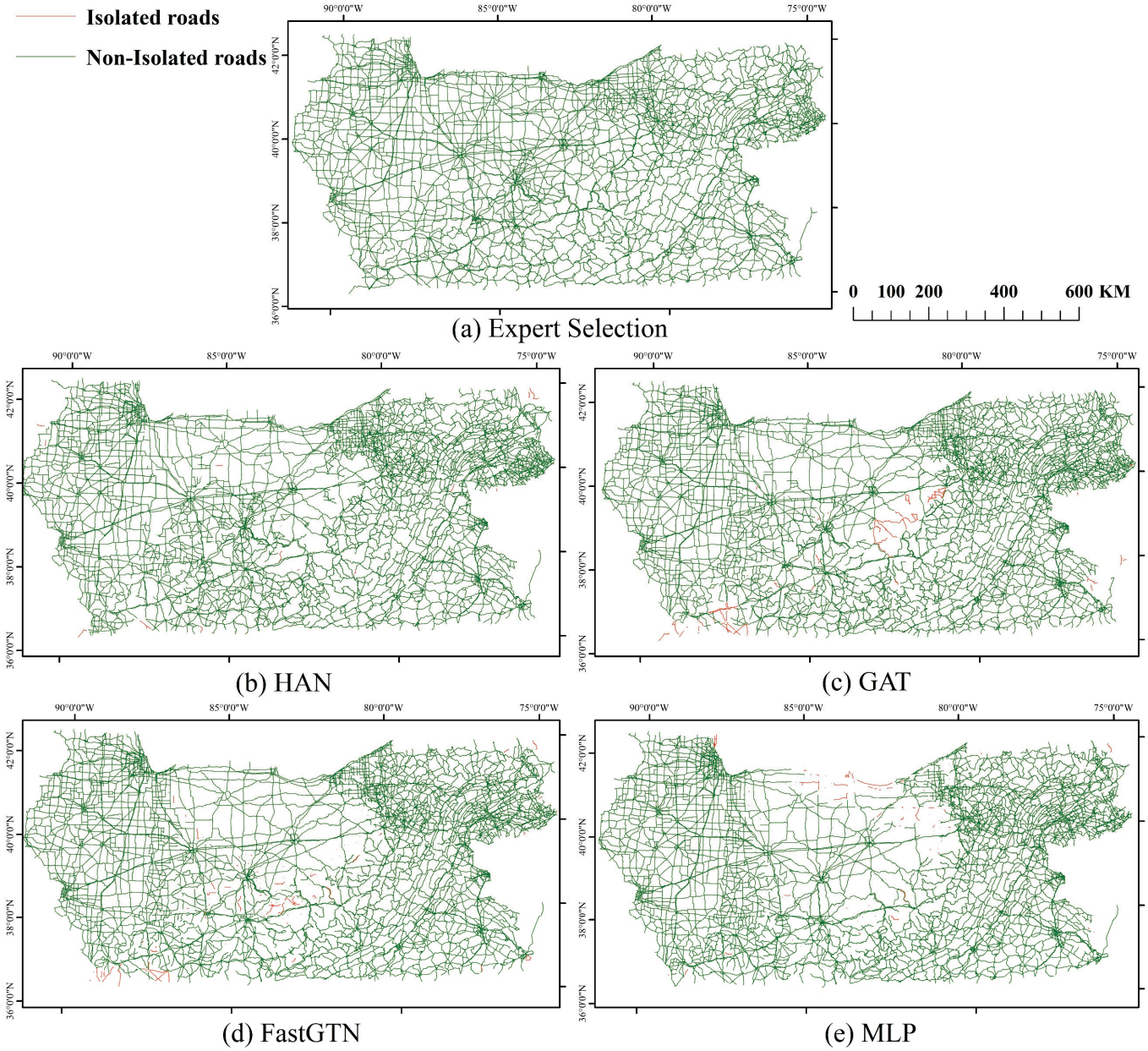


Figure 12. The isolated roads of the HAN and other models in the transductive task.

To assess the spatial equilibrium of road network density, this study partitions the road network into 5 km grids aligned with the study area. The standard deviation of road network density across these grids is computed to determine the evenness of the selected road density distribution. The experts' selected 5 km grid exhibits a road network density standard deviation of 0.059187, whereas the HAN model exhibits a standard deviation of 0.069638, the GAT model has a standard deviation of 0.069483, the FastGTN model has a standard deviation of 0.069356, and the MLP model exhibits a standard deviation of 0.073987.

To examine the spatial distribution of selection and deletion outcomes from selection models, Figure 13 compares the spatial distribution of the selection results from the four models with the expert selection results. Notably, all four selection models exhibit a significant number of erroneous deletions in the central region of the road network and a notable number of erroneous selections in the upper right corner.

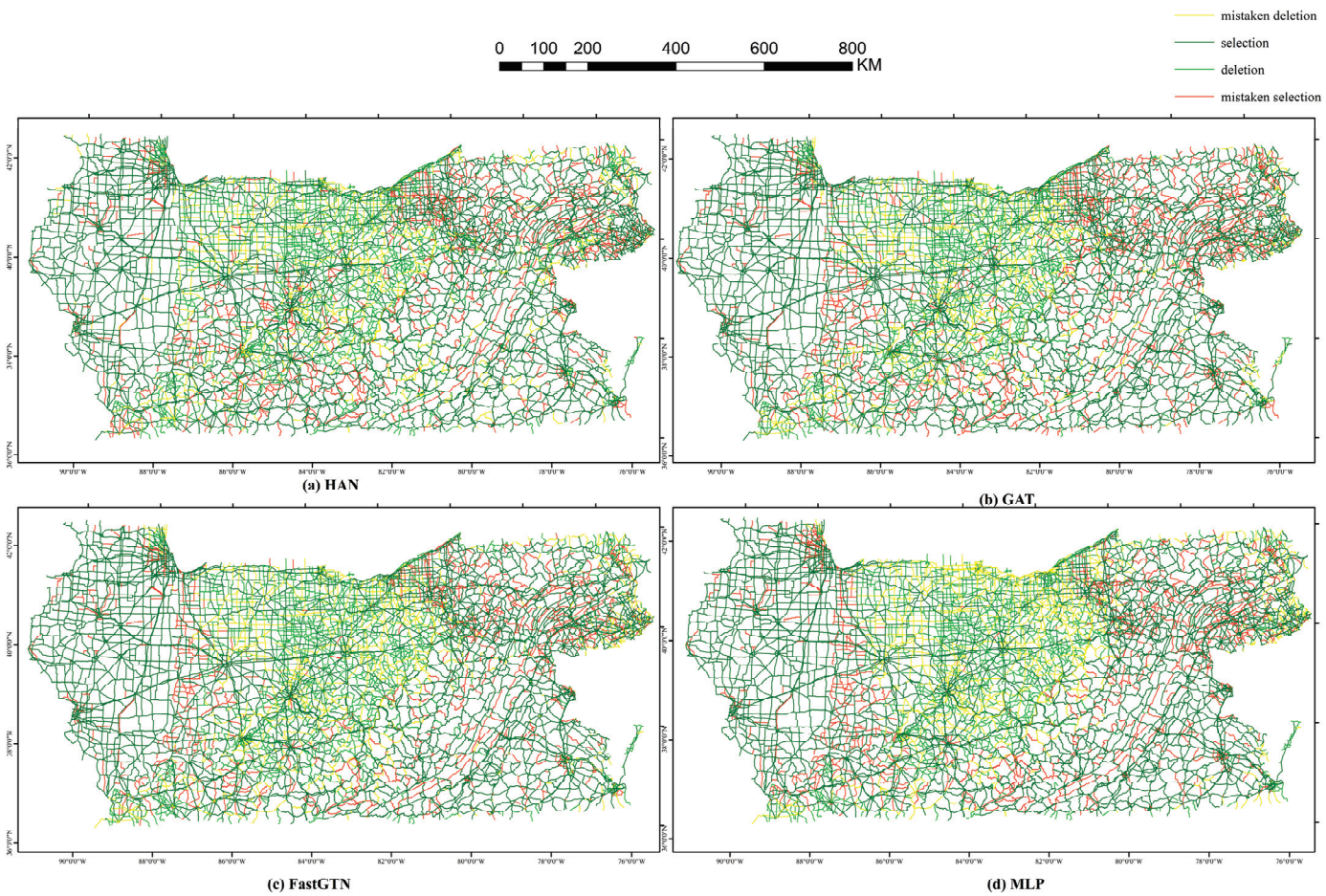


Figure 13. The spatial distribution of the selection results from different methods in the inductive task.

4.3.2. Analysis of the Road Selection Results in the Inductive Task

The inductive task involves training a model on a graph dataset with numerous known labels to predict the labels of nodes in a new graph. This task is particularly suitable for road auto-selection tasks in different spatial domains. The parameters for different models are configured as follows: The HAN model features 512 hidden layers, 4 attentional heads, and a dropout rate of 0.1, while the GAT model has 64 hidden layers, 4 attentional heads, and a dropout rate of 0.1. On the other hand, the MLP has 512 hidden layers, and the FastGTN has 512 hidden layers and 4 channels. The training step size and weight decay rate are manually adjusted. A total of 1000 epochs are trained, and the model with the highest AUC on the validation set is selected as the final model for testing on the test set. Due to the limited number of CR road types, they were not analyzed separately.

Table 9 indicates that the HAN model outperforms other models in selecting State-type roads, achieving higher accuracy (ACC) and f1-score. Specifically, the HAN model surpasses other optimal models by 0.67% and 0.81% in terms of ACC and f1-score, respectively. In terms of the quantity and total length of isolated roads, the HAN model falls short of other optimal models, with 19 fewer roads and 365.76 KM less than other models.

The comparison of the selection results of the four models with the expert selection results is depicted in Figure 14. The spatial distribution of the road network selected by the MLP model is highly dispersed, with large voids in the middle. This occurs because the MLP cannot aggregate information from neighboring nodes to update its nodes, such as a graph neural network. Consequently, a certain amount of topological neighborhood information is neglected during model training, resulting in a poor spatial distribution of the selected results. Numerous isolated roads are present in both the FastGTN and GAT models. Overall, the HAN model performs better than the other models, exhibiting

the minimum number of isolated roads. The density standard deviation of the 5 km grid selected by the experts is 0.06117, while that of the HAN model is 0.058875, the GAT model is 0.059803, the FastGTN model is 0.06099, and the MLP model is 0.072805.

Table 9. The statistical results of road selection models in the inductive task.

Evaluation Metrics	Model	All	State	US	Interstate
ACC	HAN	0.7021	0.5589	0.8306	0.7452
	GAT	0.7011	0.5522	0.8225	0.7962
	FastGTN	0.6904	0.5322	0.8339	0.7325
	MLP	0.6608	0.5038	0.7964	0.7093
F1 score for positive samples.	HAN	0.7804	0.4667	0.9050	0.8507
	GAT	0.7799	0.4586	0.9004	0.8806
	FastGTN	0.7710	0.4343	0.9068	0.8582
	MLP	0.7519	0.4000	0.8932	0.8545
Isolated roads (number | total length)	HAN	18 | 77.62 km	3 | 11.91 km	15 | 65.72 km	0 | 0 km
	GAT	89 | 695.19 km	40 | 320.10 km	48 | 374.08 km	1 | 1.02 km
	FastGTN	37 | 443.38 km	10 | 189.37 km	27 | 254.02 km	0 | 0 km
	MLP	72 | 242.31 km	2 | 2.36 km	52 | 191.15 km	18 | 48.80 km
Road network density (km/km^2)	Expert Selection	0.10662	0.03395	0.06002	0.01266
	HAN	0.11179	0.03924	0.06015	0.01361
	GAT	0.11228	0.03991	0.05938	0.01299
	FastGTN	0.11916	0.04468	0.06118	0.01330
	MLP	0.09833	0.03250	0.05368	0.01215

To examine the spatial distribution of the selection and deletion results of the four models, Figure 15 compares their selection outcomes with the expert's selection. The comparison reveals that the MLP model exhibits numerous mistakenly deleted roads concentrated in the middle of the road network, while the FastGTN model shows more mistaken selections in the road network. In contrast, the HAN and GAT models display more evenly distributed patterns of mistakenly deleted roads. These findings suggest that the HAN and GAT models exhibit more balanced and consistent selection results compared with the other models.

4.3.3. Analysis of Ablation Study Results

To gain a deeper understanding of the advantages of aggregating neighboring road features based on the meta-paths we designed, we analyzed the results of the two types of experiments from an ablation perspective. For the transductive task, as depicted in Table 8, when compared with the results obtained from our designed meta-path selection (HAN model selection), removing this meta-path selection (GAT selection) led to an increase of 158 isolated roads and a total length increase of 1414.44 km. Substituting it with the results from aggregating road neighbor node features using meta-paths of arbitrary lengths and types (FastGTN) resulted in an increase of 241 isolated roads and a total length increase of 999.0 km. Similarly, for the inductive task, as shown in Table 9, removing the meta-path selection (GAT selection) caused an increase of 71 isolated roads and a total length increase of 617.57 km. Substituting it with the results from aggregating road neighbor node features using meta-paths of arbitrary lengths and types (FastGTN) led to an increase of 37 isolated roads and a total length increase of 365.76 km. These findings demonstrate that aggregating neighboring road features according to the meta-paths we designed can enhance the connectivity of the selection results.

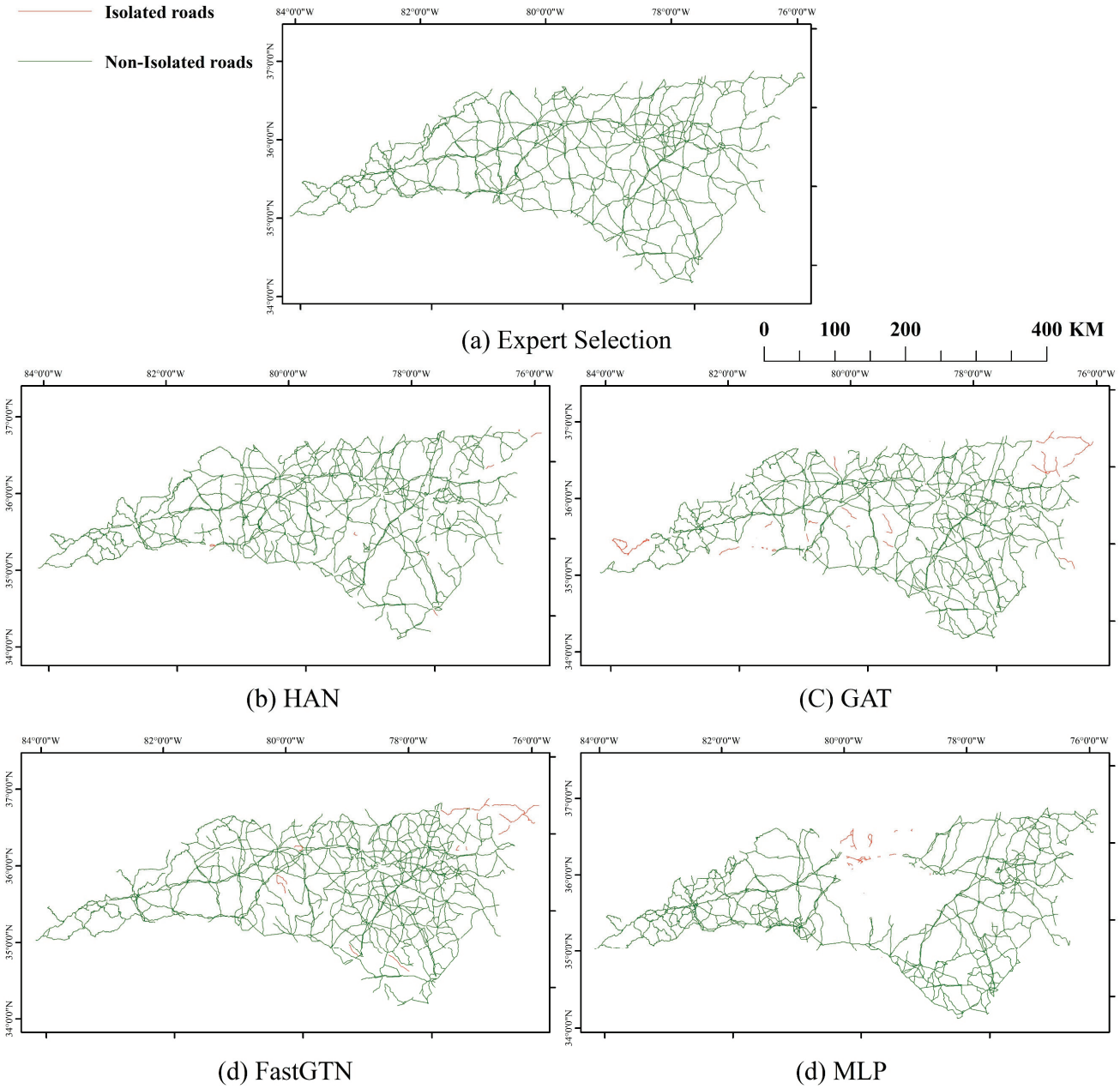


Figure 14. The isolated roads of the HAN and other models in the inductive task.

4.4. Exploring Selection Performance at Various Scales and in Various Locations

To investigate the selection performance of the HAN road network automatic selection model across different regions and scales, this paper utilizes the 1:250,000 road network data of Hebi City, China, as the foundational test dataset. Subsequently, the 1:1,000,000 road network data is selected through the application of the HAN road network selection model. Given that the road network data used in this study exhibits incomplete semantic features pertaining to road levels, this paper employs a back-propagation method based on semantic levels [53] to classify the road network levels. Furthermore, the square root model is utilized to determine the sequence of selections, proceeding from higher to lower levels. Thereafter, the HAN Road Network Automatic Selection Model is employed to select roads at a scale of 1:1,000,000, specifically targeting a particular type of road that was not fully selected previously. This process yields the selection probability of each road segment, which is ultimately divided by a threshold. The selection outcomes are presented in Figure 16, demonstrating the preservation of the spatial distribution and connectivity of

the road network. These results indicate that the HAN road network automatic selection model exhibits robust performance in selecting road networks of various scales and across different regions.

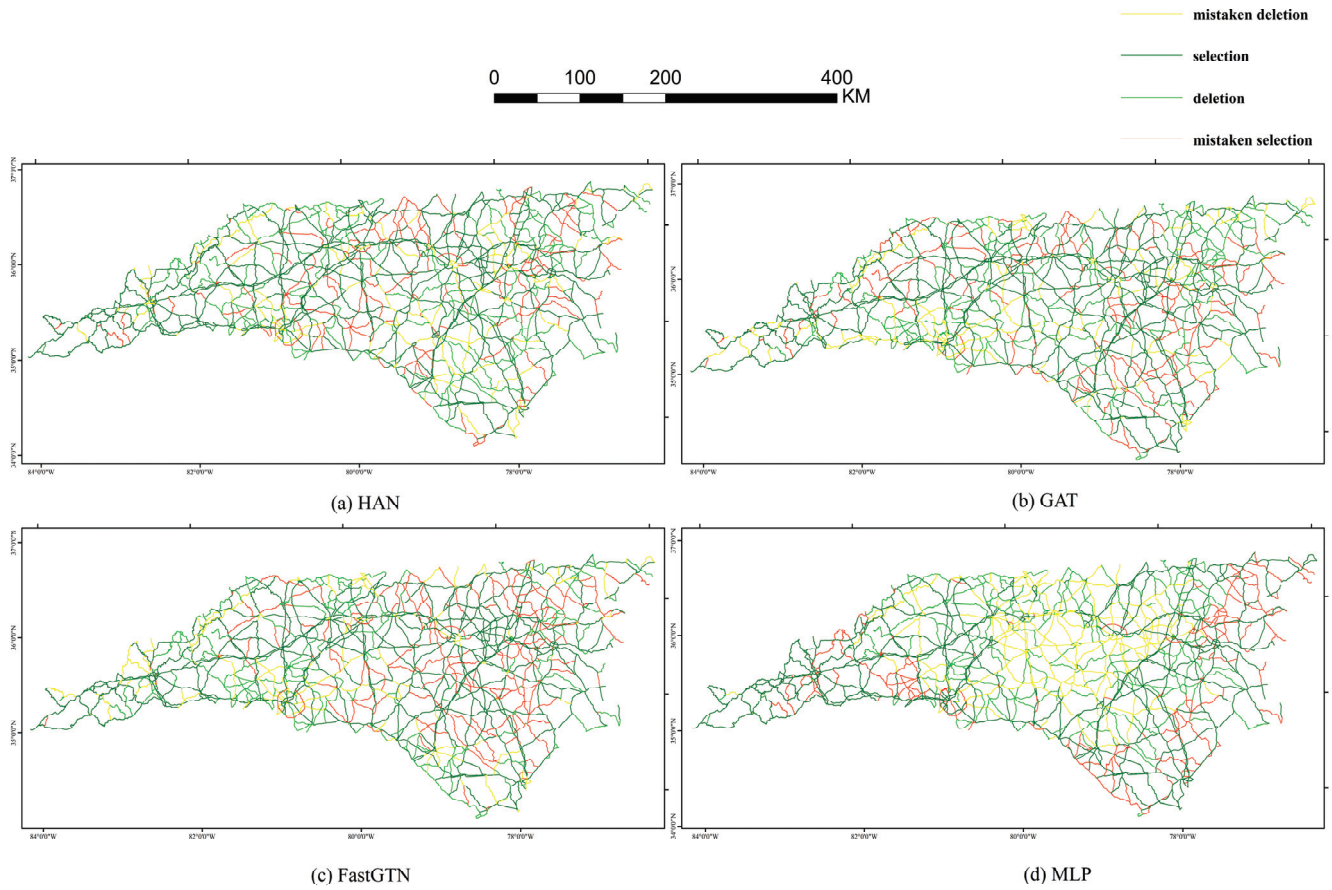


Figure 15. The spatial distribution of the selection results from different methods.

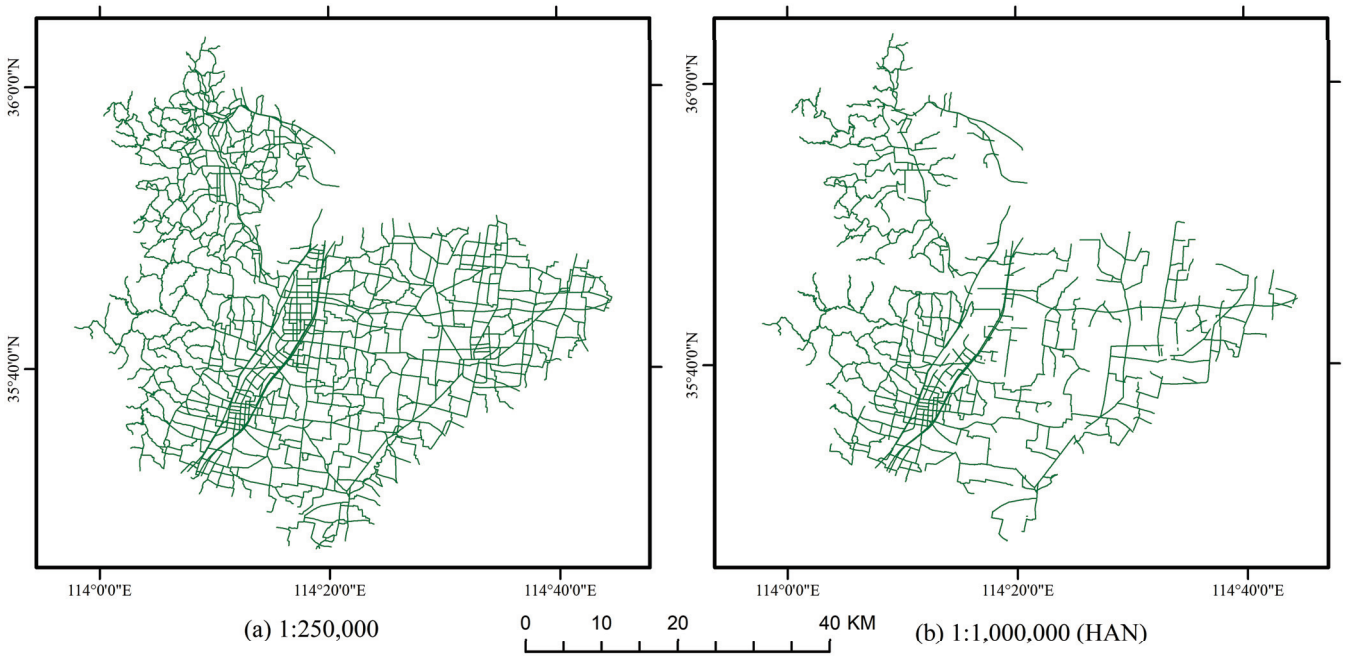


Figure 16. Results of road network selection for various regions and scales of the HAN model.

5. Conclusions and Discussion

Our objective is to tackle the challenges encountered by homogeneous graph neural networks in accurately selecting road types within a road network, especially when dealing with similar numbers of selections and deletions. Additionally, we aim to enhance the spatial distribution of the selected road network. To achieve this, we have devised two distinct meta-paths for each road type, adhering to the principle of aggregating the most pertinent features from neighboring nodes. This approach allows us to construct a heterogeneous graph attention network specifically designed for intelligent road network selection.

In addition, we have applied the feature masking method to explore the significance of three key road features in automatic road network selection models based on graph neural networks: geometric features, semantic features, and topological features. Our findings uncover that road type stands out as the most crucial feature, closely followed by start and end points (X, Y), closeness centrality, degree centrality, length, degree, curvature ratio, and betweenness centrality. However, the effects of mesh density, the number of road vertices, and eigenvector centrality on the automatic road network selection task of graph neural networks are slight.

For the transductive road network selection task, the HAN model demonstrates superior performance compared with the other optimal models. It achieves a 1.62% higher accuracy and a 1.43% higher F1-score in the selection and deletion of a similar number of State Road types. When considering the overall selection results, the HAN model further outperforms the remaining optimal models by increasing accuracy metrics and F1-Score by 1.02% and 1.01%, respectively. Additionally, the HAN model significantly reduces the number of isolated road networks and the total length of the selection results by 79.78% and 63.60%, respectively, compared with the other models.

In the inductive road network selection task, the HAN model also excels, achieving a 0.67% higher accuracy index and a 0.81% higher F1-Score than the remaining optimal models for State Road types. Furthermore, the HAN model reduces the number and length of isolated road networks by 51.35% and 67.96%, respectively, compared with the other models. This improved performance can be attributed to the second-layer convolution of the HAN model, which effectively extracts a wider range of peripheral node features, as four out of its five mating paths are second-order neighbors.

It is worth emphasizing that while FastGTN can indeed automatically generate a range of meta-paths, it falls short in effectively incorporating the background knowledge specific to road network selection. Experimental results underscore the superiority of the HAN model presented in this article, which is explicitly designed based on road correlation. When compared with the FastGTN model, the HAN model demonstrates superior efficiency and accuracy in road network selection tasks. This advantage can be attributed to the HAN model's ability to leverage domain-specific knowledge, enabling it to make more informed and precise decisions during the selection process.

The primary contribution of this paper lies in quantifying the significance of road features within graph neural network models, a set of evaluation metrics used to assess the road network selection results, and innovatively designing two distinct meta-paths for road network aggregation based on neighboring node features. By applying a heterogeneous graph neural network to the automated selection of road networks, we have established individual, automated selection models tailored for each road type.

Across two road network selection tasks, our approach enhances the precision of selecting road types with a comparable number of deletions and significantly mitigates the issue of excessively isolated roads within the selected network. Notably, our current research has not delved into the model's performance for road network selection using existing methods to generate road types in scenarios where road type attributes are absent.

Future work will focus on exploring which methods generate road types that yield optimal performance within this model. Additionally, since this paper's training utilized road network data of a single scale, our future exploration will delve into the enhancement

of road network selection performance through joint training across multiple scales., thus broadening its applicability and further validating its utility in real-world scenarios.

Author Contributions: Conceptualization, Haohua Zheng and Jianchen Zhang; methodology, Haohua Zheng and Jianchen Zhang; software, Haohua Zheng; resources, Jiayao Wang and Jianzhong Guo; writing—original draft preparation, Haohua Zheng and Jianchen Zhang; writing—review and editing, Heying Li and Guangxia Wang. All authors have read and agreed to the published version of the manuscript.

Funding: The Project was supported by the Natural Science Foundation of Henan (grant number 232300420436, 242300420589), Science and Technology Development Project of Henan Province (grant number 242102210175), National Natural Science Foundation of China (grant number U21A2014), Key Scientific Research Projects in Colleges and Universities of Henan Province (grant number 24B170002), and Henan Collaborative Innovation Center of Geo-Information Technology for Smart Central Plains (grant number 2023C001).

Data Availability Statement: The original data presented in the study are openly available in FigShare at https://doi.org/10.6084/m9.figshare.26826400 (accessed on 22 August 2024).

Acknowledgments: We would like to thank the anonymous reviewers for their insightful comments and substantial help in improving this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, Z.; Liu, W.; Xu, Z.; Ti, P.; Gao, P.; Yan, C.; Lin, Y.; Li, R.; Liu, C. Cartographic Representation of Spatio-Temporal Data: Fundamental Issues and Research Progress. *Acta Geod. Cartogr. Sin.* **2021**, *50*, 1033.
2. Wang, J.; Wu, F.; Yan, H. Cartography: Its Past, Present and Future. *Acta Geod. Et Cartogr. Sin.* **2022**, *51*, 829.
3. Reichstein, M.; Camps-Valls, G.; Stevens, B.; Jung, M.; Denzler, J.; Carvalhais, N.; Prabhat. Deep Learning and Process Understanding for Data-Driven Earth System Science. *Nature* **2019**, *566*, 195–204. [CrossRef]
4. Kronenfeld, B.J.; Buttenfield, B.P.; Stanislawski, L.V. Map Generalization for the Future: Editorial Comments on the Special Issue. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 468. [CrossRef]
5. Li, Z.; Lan, T.; Ti, P.; Xu, Z. Advances in Cartography from the Perspective of Maslow's Hierarchy of Needs. *Acta Geod. Cartogr. Sin.* **2022**, *51*, 1536.
6. Sun, Y.; Han, J. Mining Heterogeneous Information Networks: A Structural Analysis Approach. *SIGKDD Explor. Newsl.* **2013**, *14*, 20–28. [CrossRef]
7. Zhang, J.; Lu, C.T.; Zhou, M.; Xie, S.; Chang, Y.; Yu, P.S. Heer: Heterogeneous Graph Embedding for Emerging Relation Detection from News. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016.
8. Dong, Y.; Chawla, N.V.; Swami, A. Metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 135–144.
9. Sankar, A.; Zhang, X.; Chang, K.C.C. Meta-Gnn: Metagraph Neural Network for Semi-Supervised Learning in Attributed Heterogeneous Information Networks. In Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Vancouver, BC, Canada, 27–30 August 2019.
10. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; Yu, P.S. Heterogeneous Graph Attention Network. In Proceedings of the World Wide Web Conference 2019, Online, 13–17 May 2019.
11. Yun, S.; Jeong, M.; Kim, R.; Kang, J.; Kim, H.J. Graph Transformer Networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2019.
12. Zhou, S.; Bu, J.; Wang, X.; Chen, J.; Hu, B.; Chen, D.; Wang, C. Hahe: Hierarchical Attentive Heterogeneous Information Network Embedding. *arXiv* **2019**, arXiv:1902.01475.
13. Jia, X.; Dong, Y.; Zhu, F.; Qian, J. Research Progress of Heterogeneous Graph Convolutional Networks. *Comput. Eng. Appl.* **2021**, *57*, 36–49.
14. Yun, S.; Jeong, M.; Yoo, S.; Lee, S.; Yi, S.S.; Kim, R.; Kang, J.; Kim, H.J. Graph Transformer Networks: Learning Meta-Path Graphs to Improve Gnns. *Neural Netw.* **2022**, *153*, 104–119. [CrossRef]
15. Bing, R.; Yuan, G.; Zhu, M.; Meng, F.; Ma, H.; Qiao, S. Heterogeneous Graph Neural Networks Analysis: A Survey of Techniques, Evaluations and Applications. *Artif. Intell. Rev.* **2023**, *56*, 8003–8042. [CrossRef]
16. Wu, F.; Gong, X.; Du, J. Overview of the Research Progress in Automated Map Generalization. *Acta Geod. Cartogr. Sin.* **2017**, *46*, 1645–1664.
17. Jiang, B.; Claramunt, C. A Structural Approach to the Model Generalization of an Urban Street Network*. *GeoInformatica* **2004**, *8*, 157–171. [CrossRef]

18. Touya, G. A Road Network Selection Process Based on Data Enrichment and Structure Detection. *Trans. GIS* **2010**, *14*, 595–614. [CrossRef]
19. Weiss, R.; Weibel, R. Road Network Selection for Small-Scale Maps Using an Improved Centrality-Based Algorithm. *J. Spat. Inf. Sci.* **2014**, *9*, 71–99. [CrossRef]
20. Shoman, W.; Gülgen, F. Centrality-Based Hierarchy for Street Network Generalization in Multi-Resolution Maps. *Geocarto Int.* **2016**, *32*, 1352–1366. [CrossRef]
21. Gülgen, F.; Gökgöz, T. A Block-Based Selection Method for Road Network Generalization. *Int. J. Digit. Earth* **2011**, *4*, 133–153. [CrossRef]
22. Thomson, R.C. The Stroke' Concept in Geographic Network Generalization and Analysis. In *Progress in Spatial Data Handling: 12th International Symposium on Spatial Data Handling*; Riedl, A., Kainz, W., Elmes, G.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 681–697.
23. Zhou, Q.; Li, Z. A Comparative Study of Various Strategies to Concatenate Road Segments into Strokes for Map Generalization. *Int. J. Geogr. Inf. Sci.* **2012**, *26*, 691–715. [CrossRef]
24. Chen, J.; Hu, Y.; Li, Z.; Zhao, R.; Meng, L. Selective Omission of Road Features Based on Mesh Density for Automatic Map Generalization. *Int. J. Geogr. Inf. Sci.* **2009**, *23*, 1013–1032. [CrossRef]
25. Zhou, Q.; Li, Z. Empirical Determination of Geometric Parameters for Selective Omission in a Road Network. *Int. J. Geogr. Inf. Sci.* **2015**, *30*, 263–299. [CrossRef]
26. Liu, Y.; Li, W. A New Algorithms of Stroke Generation Considering Geometric and Structural Properties of Road Network. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 304. [CrossRef]
27. Benz, S.A.; Weibel, R. Road Network Selection for Medium Scales Using an Extended Stroke-Mesh Combination Algorithm. *Cartogr. Geogr. Inf. Sci.* **2014**, *41*, 323–339. [CrossRef]
28. Xu, Z.; Wang, Z.; Yan, H.; Wu, F.; Duan, X.; Sun, L. A Method for Automatic Road Selection Combined with Poi Data. *J. Geo-Inf. Sci.* **2018**, *20*, 159–166.
29. Deng, M.; Chen, X.; Tang, J.; Liu, H.; He, J. A Method for Road Network Selection Considering the Traffic Flowsemantic Information. *Geomat. Inf. Sci. Wuhan Univ.* **2020**, *45*, 1438–1447.
30. Yu, W.; Zhang, Y.; Ai, T.; Guan, Q.; Chen, Z.; Li, H. Road Network Generalization Considering Traffic Flow Patterns. *Int. J. Geogr. Inf. Sci.* **2020**, *34*, 119–149. [CrossRef]
31. Lyu, Z.; Sun, Q.; Ma, J.; Xu, Q.; Li, Y.; Zhang, F. Road Network Generalization Method Constrained by Residential Areas. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 159. [CrossRef]
32. Liu, K.; Ma, J.S. *Research on Intelligent Selection Ofroad Network Automatic Generalization Based on Kernel-Based Machine Learning*; Nanjing University: Nanjing, China, 2017.
33. Liu, K.; Li, J.; Shen, J.; Ma, J. Selection of Road Network Using Bp Neural Networkand Topological Parameters. *J. Geomat. Sci. Technol.* **2016**, *33*, 325–330.
34. Liu, P.; Yuan, L.H.; Zhang, K.; Shen, J.; Ma, J.S. Intelligent Selection of Osm Road Network Based on Rbf Neural Network. *Geomat. World* **2019**, *26*, 8–13.
35. Karsznia, I.; Wereszczyńska, K.; Weibel, R. Make It Simple: Effective Road Selection for Small-Scale Map Design Using Decision-Tree-Based Models. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 457. [CrossRef]
36. Guo, X.; Qian, H.; Wang, X.; Liu, J.; Ren, Y.; Zhao, Y.; Chen, G. Ontology Knowledge Reasonin Method for Multi-Source Intelligent Road Selection. *Acta Geod. Cartogr. Sin.* **2022**, *51*, 279–289.
37. Karsznia, I.; Adolf, A.; Leyk, S.; Weibel, R. Using Machine Learning and Data Enrichment in the Selection of Roads for Small-Scale Maps. *Cartogr. Geogr. Inf. Sci.* **2023**, *51*, 60–78. [CrossRef]
38. Wang, J.; Cui, T.; Wang, G. Application of Graph Theory in Automatic Selection of Road Network. *J. Geomat. Sci. Technol.* **1985**, *2*, 79–86.
39. Liu, G.; Li, Y.; Yang, J.; Zhang, X. Auto-Selection Method of Road Networks Based on Evaluation of Node Importance for Dual Graph. *Acta Geod. Cartogr. Sin.* **2014**, *43*, 97–104.
40. Cao, W.; Zhang, H.; He, J.; Lan, T. Road Selection Considering Structural and Geometric Properties. *Geomat. Inf. Sci. Wuhan Univ.* **2017**, *42*, 520–524.
41. Ma, C.; Sun, Q.; Chen, H.; Xu, Q.; Wen, B. Application of Weighted Pagerank Algorithm in Road Network Auto-Selection. *Geomat. Inf. Sci. Wuhan Univ.* **2018**, *43*, 1159–1165.
42. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
43. Velikovi, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2017**, arXiv:1710.10903.
44. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017.
45. Courtial, A.; Touya, G.; Zhang, X. Can Graph Convolution Networks Learn Spatial Relations? Abstract of ICA. In Proceedings of the 30th International Cartographic Conference, Florence, Italy, 14–18 December 2021; Volume 3, pp. 1–2.
46. Zhang, K.; Zheng, J.; Shen, J.; Ma, J. Application of the Graph Convolution Network in the Selection of Road Network. *Sci. Surv. Mapp.* **2021**, *46*, 165–170+177.
47. Zheng, J.; Gao, Z.; Ma, J.; Shen, J.; Zhang, K. Zhang. Deep Graph Convolutional Networks for Accurate Automatic Road Network Selection. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 768. [CrossRef]

48. Ma, C.; Xiong, S.; Jiang, D. Application of the Graph Convolution Network in the Road Network Auto-Selection. *Sci. Surv. Mapp.* **2022**, *47*, 200–205+215.
49. Zhu, Y.; Yang, M.; Yan, X. A Road Network Selection Method Using Graph Convolutional Network. *Beijing Surv. Mapp.* **2022**, *36*, 1455–1459.
50. Guo, X.; Liu, J.; Wu, F.; Qian, H. A Method for Intelligent Road Network Selection Based on Graph Neural Network. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 336. [CrossRef]
51. Wang, D.; Qian, H. Graph Neural Network Method for the Intelligent Selection of River System. *Geocarto Int.* **2023**, *38*, 2252762. [CrossRef]
52. Yu, H.; Ai, T.; Yang, M.; Li, J.; Wang, L.; Gao, A.; Xiao, T.; Zhou, Z. Integrating Domain Knowledge and Graph Convolutional Neural Networks to Support River Network Selection. *Trans. GIS* **2023**, *27*, 1898–1927. [CrossRef]
53. He, H.; Qian, H.; Liu, H.; Wang, X.; Hu, H. Road Network Selection Based on Road Hierarchical Structure Control. *Acta Geod. Cartogr. Sin.* **2015**, *44*, 453–461+470.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

geomatics

MDPI

Article

Deep Learning for Urban Tree Canopy Coverage Analysis: A Comparison and Case Study

Grayson R. Morgan *, Danny Zlotnick, Luke North, Cade Smith and Lane Stevenson

Department of Geography, Brigham Young University, Provo, UT 84606, USA; dzlotnic@student.byu.edu (D.Z.); lnorth01@student.byu.edu (L.N.); lbs44@student.byu.edu (L.S.)
* Correspondence: grayson_morgan@byu.edu

Abstract: Urban tree canopy (UTC) coverage, or area, is an important metric for monitoring changes in UTC over large areas within a municipality. Several methods have been used to obtain these data, but remote sensing image classification is one of the fastest and most reliable over large areas. However, most studies have tested only one or two classification methods to accomplish this while using costly satellite imagery or LiDAR data. This study seeks to compare three urban tree canopy cover classifiers by testing a deep learning U-Net convolutional neural network (CNN), support vector machine learning classifier (SVM) and a random forests machine learning classifier (RF) on cost-free 2012 aerial imagery over a small southern USA city and midsize, growing southern USA city. The results of the experiment are then used to decide the best classifier and apply it to more recent aerial imagery to determine canopy changes over a 10-year period. The changes are subsequently compared visually and statistically with recent urban heat maps derived from thermal Landsat 9 satellite data to compare the means of temperatures within areas of UTC loss and no change. The U-Net CNN classifier proved to provide the best overall accuracy for both cities (89.8% and 91.4%), while also requiring the most training and classification time. When compared spatially with city heat maps, city periphery regions were most impacted by substantial changes in UTC area as cities grow and the outer regions get warmer. Furthermore, areas of UTC loss had higher temperatures than those areas with no canopy change. The broader impacts of this study reach the urban forestry managers at the local, state/province, and national levels as they seek to provide data-driven decisions for policy makers.

Keywords: urban forestry; urban tree canopy; remote sensing; deep learning; machine learning; aerial imagery; land cover classification

1. Introduction

As urban land in the United States continues to expand, urban forests will continue to play a substantial role in the overall health of urban and rural areas alike [1,2]. Urban forests, defined as individual or clusters of trees within or adjacent to an urban setting, provide several well-documented economic, ecological, and health benefits for residents of the urban environments [3,4]. For example, urban trees provide shade and mitigate the effects of the urban heat island effect [5,6]. They also combat particle air pollution and CO_2 emissions by sequestering carbon [7,8], are linked to positive effects in human psychology in urban environments [9–11], and reduce stormwater runoff [12,13]. Despite their enormous benefits, urban tree canopy (UTC) coverage is in decline as considerable urban growth in some regions has necessitated the removal of urban forest, leaving some cities struggling with the urban heat island (UHI) effect and other challenges [2]. Managing UTC necessitates accurate measurement and inventory of urban trees within a city, and many municipalities within the Unites States of America (USA) have created organizations or branches of government to oversee the changing urban landscapes [14].

While tagging and inventorying trees in the field creates a valuable and accurate catalog, the process consumes significant time and monetary resources. Many have turned

to less time-intensive methods, such as using remote sensing technologies, to provide a review of the urban tree status [15,16]. Remote sensing technologies, such as light detection and ranging (LiDAR), unoccupied aerial vehicles (UAVs), and aerial and satellite imagery, have been used to inventory individual trees or tree canopy coverage in towns and cities [17,18]. LiDAR data provide accurate representation of tree canopies, but can be costly and require significant computational resources for processing large datasets [19–21]. UAV data are captured in very high spatial resolutions that is effective in identifying trees and tree canopy, but it is difficult to capture data large spatial extents, i.e., an entire municipality [22–24]. Satellite and aerial imagery are being increasingly used in the context of mapping urban tree canopies because of the increasingly high spatial resolution and wide availability [25–28].

Recent advancements in computer science have increased the applicability of both machine learning and deep learning image classifiers for remote sensing applications [29–31]. State-of-the-art machine and deep learning models are available to UTC managers to quickly obtain accurate information regarding tree canopy coverage and how the coverage is changing over time. Accuracy is of the utmost importance as these data are used for legislative decision making and funding [14]. These models are now being offered as tools in accessible and easy-to-use ways through popular GIS software, such as ArcGIS Pro. Recently, scientists have found it possible to detect individual trees or calculate individual tree canopies within UTC coverage for large areal extents with both machine learning and deep learning classifiers and detectors. Yan et al. [32] compared convolutional neural networks (CNNs), support vector machine (SVM) and random forests (RF) models for UTC mapping and found CNNs performed the best. However, they used expensive satellite imagery and investigated delineating individual tree canopies in a city instead of total UTC area within an urban environment. Lv et al. [33] compared several CNNs, including one developed by their own team, but once again sought individual tree segmentation and failed to include machine learning classifiers in their comparison. Zamboni et al. [34] likewise surveyed a large number of deep learning models for individual tree detection but did not use machine learning classifiers. Wang et al. [35] compared a U-Net deep learning model to object-based classification and found U-Net to be a superior classifier for UTC coverage area, but again did not compare the results to machine learning classifiers. Despite the successes of machine and deep learning classifiers on their own, further investigation on the worth of deep learning and machine learning from a practical perspective is important. Deep learning classifiers can require more time to train and classify than machine learning classifiers, suggesting that small differences in overall accuracy may not be worth the longer processing times [36]. Application-based studies, like [37], have created a comprehensive set of UTC data for the entire state of Wisconsin using machine learning algorithms, but the results were not compared with deep learning algorithm results. In many of the application studies, multispectral satellite data or LiDAR data are combined with aerial imagery to enhance the classification [27,28,38,39]. Therefore, comparisons with less complex and still reliable machine learning classification models, like SVM and RF, are needed on simple optical imagery readily available from a management and policy-making perspective.

The present literature on urban tree canopy mapping and UHI have focused on highly populated urban centers at the cost of neglecting smaller or mid-sized urban centers. This study is the first of its kind to compare three machine and deep learning classifiers, U-Net deep learning, support vector machine, and random forests machine learning, for mapping UTC coverage in two cities of different size and shape in the USA from the perspective of an urban tree forestry manager. From this perspective, cost, computational time, and accuracy are the most important considerations. For this purpose, free National Agriculture Imagery Program (NAIP) imagery datasets are used to provide an alternative to the costly satellite, LiDAR, and UAV data that many others use for UTC managers. Commonly used ArcGIS software and desktop computer specifications are used for analysis to simulate the resources available to urban forestry managers. Following the comparison, a change detection analysis is presented as a case study to reveal where and how tree canopies in

the two diverse cities of interest, Laurel, Mississippi (MS) and Georgetown, Texas (TX), USA, have changed over a 10-year period. Temperature maps derived from Landsat 9 thermal bands are also used to reveal the location of high temperatures in relation to tree canopy gain or loss. Such land surface temperature mapping and subsequent comparisons with tree canopy coverage is common among research projects as many seek to apply the technology to identify or solve many concerns in urban areas. In fact, several studies are now working to model changes of LST in the future using data from the past now that we have sufficiently large data histories [40,41]. Many are working hard on specific applications in large cities around the world, but few are investigating smaller cities [42–44]. We aim to expand the literature in this area as well by looking at smaller regions and compare with deep learning data, which has not yet been conducted.

We hypothesize that deep learning, as available through accessible and widely used GIS software packages, will provide the best overall accuracy results and suggest that NAIP data alone is sufficient to provide accuracies similar to other studies that use multiple datasets. Furthermore, we suggest that tree canopy loss will be more prevalent over a 10-year period for Georgetown, TX because of its rapid growth and it will be visually correlated with high temperatures in the regions of canopy loss. Section 2 of this article will introduce the study areas that make this study unique and discuss methods for classifying data and generating temperature maps. Section 3 will showcase the results of this study, while Section 4 will discuss the strengths of our approach as well as limitations. Finally, Section 5 concludes by sharing the most important information.

2. Materials and Methods

2.1. Study Areas

This study identified two cities of interest to investigate urban tree canopy coverage in the United States (Figure 1). The first of these two cities, Laurel, Mississippi, USA, represents an older and smaller urban environment. Laurel is a town of just over 17,000 people in southern Mississippi and was established as a lumber town in 1882 [45]. Situated in a humid subtropical climate, mean high temperatures can rise as high as 36 degrees Celsius in July and August and fall as low as 23.8 degrees Celsius in January [46]. Recently, Laurel has been the host of the popular home renovation television show Home Town on Home and Garden Television [47]. It has also been a designated "Tree City, USA" for over 40 years now, the first city in the state of Mississippi with that title [48]. Laurel's geographic boundaries are fairly uniform in nature, without fragmented annexes and appendages, and encircle 40.8 km^2 of city, airport, and surrounding industrial zones and forest.

Georgetown, Texas, a suburb of Austin, is representative of a mid-sized city. It has a population of 75,420. In 2022, Georgetown was the fastest-growing city by percent change with an impressive population increase rate of 14.4% [49]. Known for being a warm city, Georgetown is also in a humid subtropical climate where mean high temperatures range from 25.5 degrees Celsius to 36 degrees Celsius. The city area is a fragmented, complex, and unique 156.96 km^2. Within city boundaries, special trees titled "Heritage trees" are protected and require a permit to prune or remove [50]. However, this is a small portion of the total UTC within the city and unlike Laurel, Georgetown is not designated as a "Tree City, USA".

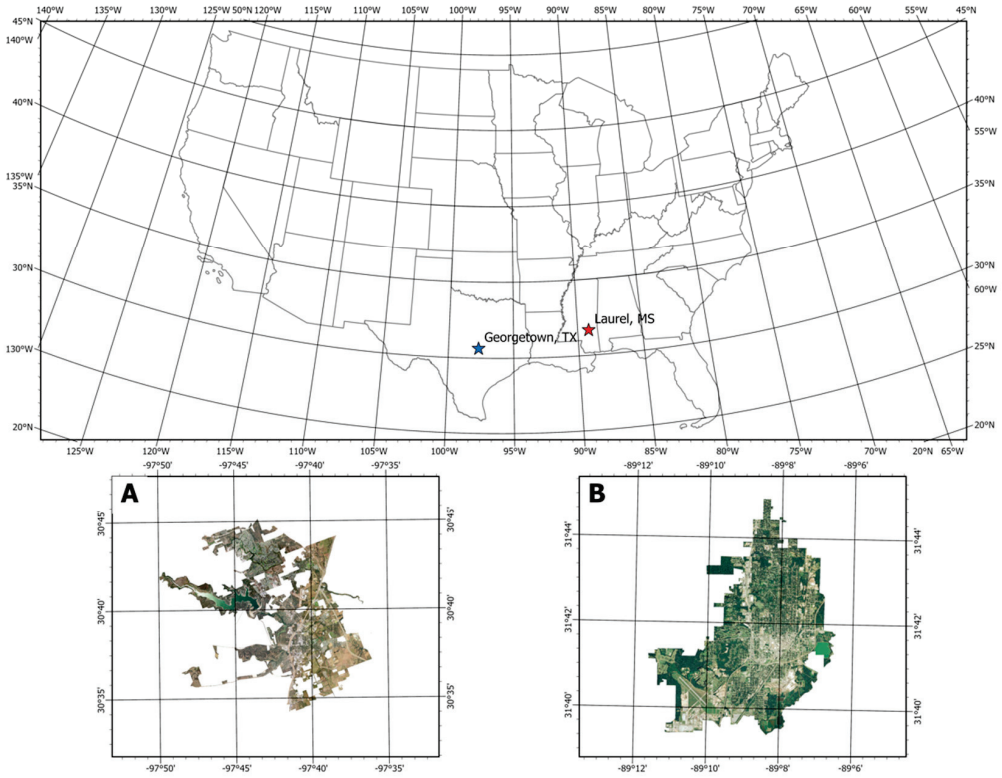


Figure 1. Study areas (**A**) Georgetown TX and (**B**) Laurel MS within the United States of America.

2.2. Remote Sensing Data

2.2.1. NAIP Imagery Data

The National Agriculture Imagery Program (NAIP) began in 2002 and is administered by the U.S. Department of Agriculture (USDA) Farm Service Agency. The goal of the program is to collect aerial imagery during leaf-on conditions. NAIP imageries are collected using digital sensors placed on aircraft that meet rigid calibration specifications [51]. NAIP imagery is generally collected at a 1 m spatial resolution and as true color (RGB) imagery, though more recent imagery has increased spatial resolutions (50–60 cm) and is additionally presented as color-infrared (CIR) imagery. NAIP imagery flights are conducted at variable altitudes with a variety of aircraft and minimum side overlap of 30%. Data management, georegistration, and quality control are all managed by the imagery program. NAIP imagery has been used for urban tree canopy in several other studies [14,37,52].

For this study, four NAIP imagery datasets were acquired and masked to each municipality's boundaries for further processing. Laurel, Mississippi datasets were acquired for 2012 and 2023, while Georgetown, TX data were gathered for 2012 and 2022. Images for Laurel were transformed into NAD83 (2011) Universal Transverse Mercator (UTM) Zone 16N, and Georgetown images were transformed into NAD83 (2011) UTM Zone 14N for further processing (Table 1).

Table 1. NAIP image data characteristics.

	2012 Georgetown Image	**2022 Georgetown Image**	**2012 Laurel Image**	**2023 Laurel Image**
Dates	25 June 2012	11 June 2022	22 August 2012	11 October 2023
Pixel size	1 m	60 cm	1 m	60 cm
Flight altitude	9052 m	4875 m	9144 m	4470 m
Aircraft	Cessna 441	Cessna Conquest and Cessna 414	Cessna Conquest	Cessna 441
Sensor	Leica Geosystems ADS80/SH82 digital sensors	Leica ADS100 pushbroom sensor	Integraph Digital Mapping Camera	Leica Geosystem's ContentMapper digital sensors
Masked image size	1.95 gb	30.15 gb	251 mb	698 mb

2.2.2. LANDSAT Data

Landsat 9 is the most recent iteration of the LANDSAT earth observation missions from NASA, having launched 27 September 2021. Landsat 9 collects a new image at each location on Earth every 16 days, resulting in the potential for cloud-free imagery at regular intervals. Cloud free Landsat tiles were collected for each city of interest on a date as close to the NAIP acquisition as possible. Unfortunately, cloudy images were often found in the image acquisition dates directly adjacent to the NAIP data acquisition dates. Nevertheless, imagery captured at dates as close to ideal were gathered and processed. The Georgetown image was gathered from 28 May 2022 and the Laurel LANDSAT image was captured 09/23/2023. The Landsat 9 thermal band 10 was predominately used, along with bands 4 (red) and 5 (NIR) for NDVI and data processing. Band 10 is collected by the Thermal Infrared Sensor 2 (TIRS-2), whose center wavelength is 10,800 nm and spatial resolution is 100 m. The band 10 image was resampled to 30 m for the purpose of this study.

After acquiring the Landsat 9 tiles from the Unites States Geological Survey EarthExplorer website (https://earthexplorer.usgs.gov, accessed on 1 April 2024), the large images were clipped to the extents of each city. Further processing was needed to create surface temperature maps using bands 4, 5, and 10. To convert the raw values into surface temperature, several steps were required. First, raw values were converted to top of atmosphere (TOA) spectral radiance. This was achieved using Equation (1):

$$\text{TOA (L)} = M_L * Q_{cal} + A_L \quad (1)$$

where M_L is the radiance multiplication factor for Band 10 found in the metadata of Landsat 9 images, Q_{cal} is the band itself, and A_L is the radiance addition factor found in the metadata as well.

After calculating TOA, we can calculate the brightness temperature from the following Equation (2):

$$\text{BT} = (\text{K2}/(\ln(\text{K1}/\text{L}) + 1)) - 273.15 \quad (2)$$

where K1 is calibration constant 1 provided in the metadata, K2 is calibration constant 2 which is also provided in the metadata, and L is the spectral radiance or TOA layer in (Watts/($m^2 * sr * \mu m$)). Subtracting 273.15 converts the results from degrees kelvin to degrees Celsius.

To improve the temperature map, further steps are needed. First, the normalized difference vegetation index (NDVI) is required to continue and is calculated from Bands 4 and 5 using the traditional equation for NDVI, found here as Equation (3):

$$\text{NDVI} = \text{float}(\text{Band 5} - \text{Band 4})/\text{float}(\text{Band 5} + \text{Band 4}) \quad (3)$$

where Band 5 is the NIR band and band 4 is the red band on Landsat 9. The NDVI is then used as an input to find the proportion of vegetation (Pv). Pv is calculated using the previously calculated NDVI band in Equation (4):

$$\text{Pv} = ((\text{NDVI} - \text{NDVImin})/(\text{NDVImax} + \text{NDVImin})) \quad (4)$$

Emissivity is then calculated based on the Pv just obtained using the equation:

$$\text{E} = 0.004 * \text{Pv} + 0.986 \quad (5)$$

Finally, land surface temperature (LST) can be calculated using several of the variables just obtained for our area of interest using the Landsat 9 bands. The final LST band was calculated using Equation (6):

$$\text{LST} = (\text{BT}/(1 + (0.00115 * \text{BT}/1.4388) * \ln(\text{E}))) \quad (6)$$

where BT is the brightness temperature layer, and E is the emissivity layer. The final result was a LST layer in degrees Celsius where we could identify spatial correlations between heat and tree canopy loss.

To assess the quality of the LST products, quality assessment (QA) data were included in the surface temperature data download from EarthExplorer. In this raster band, each pixel represents the level of uncertainty in the temperature values. This band was overlayed on the temperature raster results to indicate where higher levels of temperature uncertainty exist. In Figure 2, the areas of higher uncertainty have a pink tint and stands out against the LST map background. We found that the centers of the cities were the least affected, and the most affected regions are where water bodies existed. The water regions were removed from analysis prior to classification and case study analysis. Upon assessment, the temperature looked to be fairly underestimated, but spatial patterns were consistent with what was expected so the maps were used for analysis.

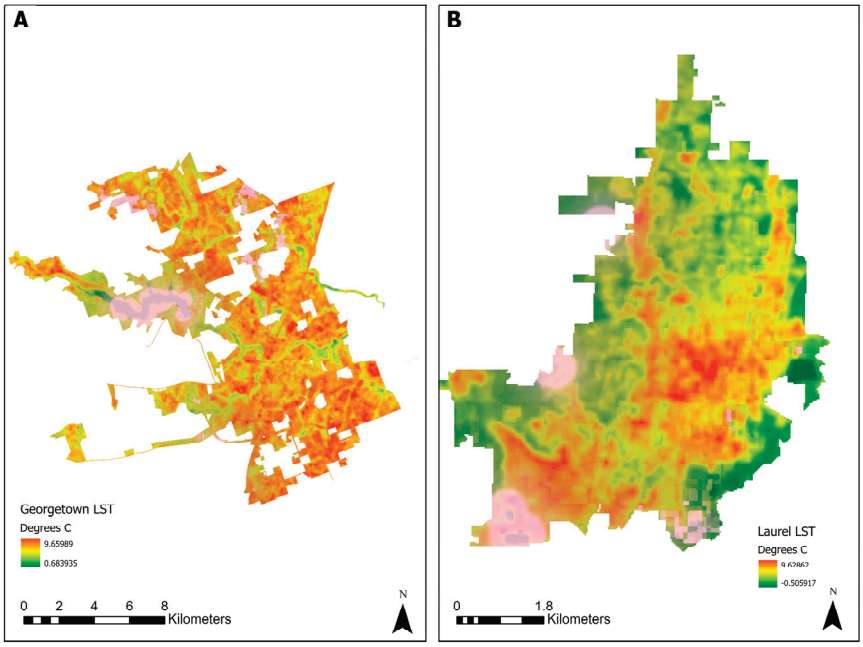


Figure 2. Land surface temperature (LST) maps for Georgetown (**A**) and Laurel (**B**) and the quality assessment overlay in pink. The pink colors indicate regions where the uncertainty is higher.

2.3. Classification and Accuracy Assessment

A general workflow of the experiment and case study analysis is presented in Figure 3. Three remote sensing image classifiers are compared, and the best performing classifier is used to classify more recent imagery, eventually resulting in a post-classification change detection and surface heat and tree canopy spatial correlation analysis.

2.3.1. Classifiers

Two traditional machine learning classifiers and a commonly used deep learning classifier are compared in this study by applying each to the same datasets with the same training data on the 2012 imagery collected for both Laurel and Georgetown. The U-Net CNN is one of the most commonly used deep learning image classifiers found in GIScience and is easy to access for UTC mangers through ESRI ArcGIS Pro 3.1.3 [53]. Originally designed for biomedical image segmentation, its applications in remote sensing image analysis are broad but now include pixel-level image classification [54]. U-Net gets its name from its U-shaped design—the architecture of the network is divided into an encoding and decoding side. U-Net has been previously used for individual tree crown extraction [35,55] and with LiDAR Data and multispectral imagery for UTC coverage mapping from UAV

data [23]. It has also shown success in modified versions of U-Net for mapping UTC [25] and other pixel-based classifications in a variety of environmental applications [36,56,57].

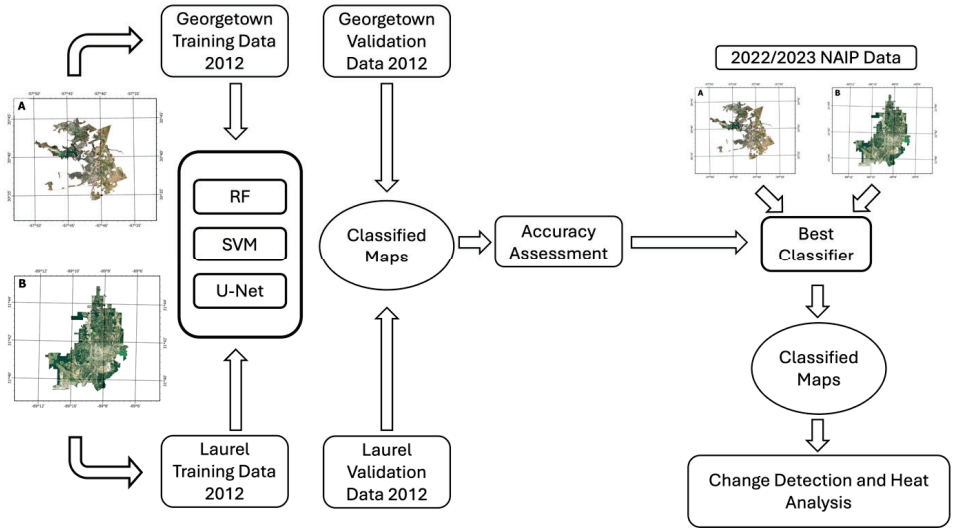


Figure 3. A general workflow of the experiment and case study.

The two traditional machine learning-based classifiers used for this study are support vector machine (SVM) and random forest (RF) classifiers. They are frequently found in the literature as high-performing image classification models ([58], p. 337). SVM is based on statistical learning theory and can be used for both linear and nonlinear classification, regression, and a series of other tasks. In its simplest form, SVM strives to find a hyperplane between the classes (it can handle many) in the target feature to separate them. Additionally, SVM classifiers do not need to have normally distributed or large training samples to perform well [59]. For this study, we used the support vector machine classifier with the maximum number of samples per class set as 1000. The segment attributes used for classification were active chromaticity color, mean digital number, and compactness.

Random forests (also called random trees in ArcGIS Pro) is also a supervised classification method that relies on a series of decisions being made based on the statistical characteristics of the training dataset and image. The decisions all stem from past decisions to make what looks like a tree when represented graphically. During the process of classification, many decision trees are made (i.e., a forest is grown) and the most frequently created output is used for the final classification [58]. Of the three classifiers used for this comparative study, the 'best' classifier will be determined by overall accuracy (OA), kappa coefficient, processing time, and ease of use. For the RF classifier, we used 150 as the maximum numbers of trees, 60 maximum tree depth, and 1000 maximum number of samples per class. The segment attributes used for classification once again were active chromaticity color, mean digital number, and compactness.

2.3.2. Classification

The same 2012 NAIP orthomosaic images were classified three times by each classifier for both Laurel and Georgetown. The NAIP imagery for 2012 clipped was to the municipality boundaries (Figure 4). Training samples and validation samples were first collected using ArcGIS Pro 3.1.3 for each of the study areas using the high resolution NAIP imagery and prior knowledge of each city. In order to increase our focus on obtaining the most accurate results of UTC area and keep classifications simple, only three classes were specified for the classifications. After testing a variety of classes, the most effective and efficient classification occurred with a tree class, grass class, and urban/barren class. Prior to classification, water bodies within municipality boundaries were masked and removed. Because NAIP imagery was clipped to municipality boundaries, only the three classes

were present and relevant to the study at hand. A total of 156 total training polygons were collected for Laurel before training the classifier, representing a total area of 1.06 km^2. A total of 267 samples were obtained for Georgetown as well, resulting in a total area of 1.41 km^2. All samples were randomly obtained in a variety of regions around the study areas in order to fairly represent the diversity of grasses, urban structures, and trees found throughout the cities.

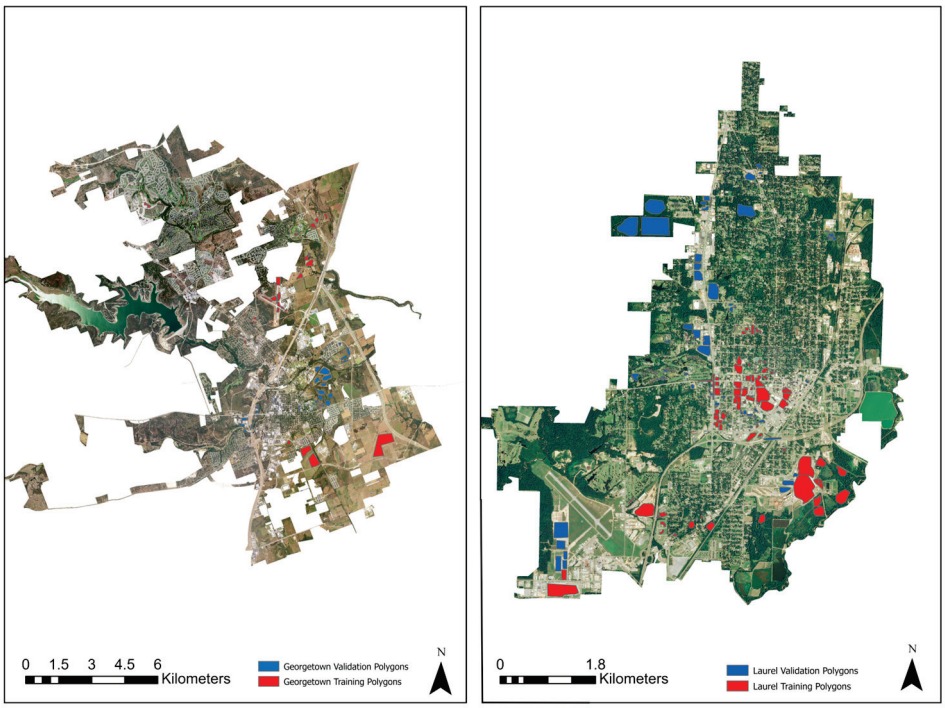


Figure 4. Training and validation samples used for classification and accuracy assessment of the 2012 NAIP images (water included for map purpose only).

The collected training polygons were once again used in ArcGIS Pro 3.1.3 to train each of the U-Net CNN, SVM, and RF classifiers. Each classification was conducted within the ArcGIS Pro graphical user interface environment to replicate the experience urban forestry managers would have when attempting similar remote-sensing based inventories.

2.3.3. Accuracy Assessment

Accuracy was assessed for the 2012 and 2022/23 classifications using 182 validation samples for Laurel and 282 polygons for Georgetown collected separately from the training polygons. No overlap between the two samples were used to ensure fair assessment (Figure 4). Four remote sensing image classification accuracy statistics, users' accuracy, producer's accuracy, overall accuracy (OA), and kappa coefficient, were used to assess each classification. The producer's accuracy is the total number of pixels classified correctly for a class based on the validation data, divided by the total pixels for that class. The user's accuracy is the number of correctly classified pixels divided by the number of pixels classified into any particular class. All four statistics were seamlessly calculated using the ArcGIS Pro 3.1.3 compute confusion matrix tool.

The confusion matrix tool was used to first generate a stratified random sample of points dispersed equally throughout the validation polygons. Then fields in the accuracy assessment points were populated from the ground truth, or validation data, as well as the classified raster. Once the fields were complete, the tool computed the OA using the following equation:

$$\mathrm{OA} = \frac{\sum_{i=1}^{K} xii}{N} \tag{7}$$

where x_{ii} represents a pixel classified correctly, N is the total number of pixels [60]. The kappa coefficient was likewise computed using the following equation:

$$\hat{K} = \frac{N\sum_{i=1}^{K} xii - \sum_{i=1}^{K}(x_{i+} \times x_{+j})}{N^2 - \sum_{i=1}^{k}(x_{i+} \times x_{+j})} \quad (8)$$

where N is the total number of samples, k is the number of rows in the confusion matrix, x_{ii} is the number of observations in row i and column i, and x_{i+} and x_{+j} are the marginal totals for row i and column j [61]. The results were presented in tables for the 2012 images and assessed to determine the best classifier. The best classifier was then used on the 2022 and 2023 images.

2.4. Case Study and Heat Mapping

In order to assess the tree canopy changes and potential impact on urban heat, the most accurate classifier was used to create an urban tree canopy map of Georgetown and Laurel in the years 2022 and 2023, respectively. New training and validation samples were digitized in ArcGIS Pro 3.1.3 by scientists familiar with the urban areas using the NAIP imagery. Once again, three classes, urban, tree canopy, and grass, were used for classification because of our specific interest in the tree canopy area. The classified images were used in a post-classification change detection, one of the most common change detection methods [62,63]. This is where two remote sensing images, representing T1 (Time 1) and T2 (Time 2), are classified into land cover classes first and one is subsequently subtracted from the other to show a change in a particular class over time. For this study, we were interested in the canopy change over the 10–11 year period between NAIP data collection dates.

The pixels representing changes in canopy, either from canopy to grass or canopy to urban/barren, were symbolized and overlaid onto the Landsat 9 heat maps generated previously. Using visual techniques described in [60], spatial correlations were identified and will be presented hereafter. To quantitatively assess temperature within the areas of canopy loss and the rest of the city boundaries, 1000 stratified random points were generated for both the areas of canopy loss and no or little change (500 points in each). Using the extract values to points ArcGIS tool, we obtained temperature values for each of the points and then compared the means of the two groups using SPSS software. The results of a independent samples *t*-test indicates if there is a statistically significant difference between the temperatures found in the areas of canopy change and the general areas of the area otherwise specified.

3. Results

3.1. Laurel, MS Classification Experiment

The Laurel, MS image map from 2012 was classified successfully using the U-Net CNN, SVM, and RF classifiers. The U-Net classifier took the most time, totaling 53 min and 11 s for both model training and classification. The SVM required a total of 8 min and 17 s, while the RF classifier only needed 4 min and 54 s. This was completed on a Dell Inspiron 5680 6-core Intel i7 central processing unit (CPU) desktop with 16 GB RAM, and a Nvidia GTX 1060 3 GB graphical processing unit (GPU). Classification times, OA, and kappa coefficient are reported in Table 2.

Table 2. Laurel, MS classification accuracy assessment.

	U-Net	**SVM**	**RF**
Overall accuracy	91.4%	84.2%	76.2%
Kappa coefficient	0.871	0.763	0.644
Classifier training time	20 m 7 s	2 m 4 s	1 m 30 s
Classification time	33 m 4 s	6 m 13 s	3 m 24 s

Despite taking longer, the overall accuracy and kappa coefficients of the U-Net classifier outperformed the same metrics from the SVM and RF classifiers. Overall accuracy of the U-Net classifier reached 91%, while SVM was adequate at 84% and RF came in third at 76%. Tables 3–5 reveal more detailed results pertaining to the users and producers' accuracy for each classifier.

Table 3. U-Net classifier confusion matrix for Laurel, MS.

	Grass	Developed	Forest	Total	User's Accuracy
Grass	382	3	0	385	99%
Developed	18	397	82	497	80%
Forest	0	0	318	318	100%
Total	400	400	400	1200	
Producer's accuracy	96%	99%	80%		

Table 4. SVM classifier confusion matrix for Laurel, MS.

	Grass	Developed	Forest	Total	User's Accuracy
Grass	286	1	13	300	95%
Developed	0	396	59	455	87%
Forest	114	3	328	445	74%
Total	400	400	400	1200	
Producer's accuracy	71.5%	99%	82%		

Table 5. RF classifier confusion matrix for Laurel, MS.

	Grass	Developed	Forest	Total	User's Accuracy
Grass	205	1	15	221	93%
Developed	97	392	68	557	70%
Forest	98	7	317	422	75%
Total	400	400	400	1200	
Producer's accuracy	51%	98%	79%		

After close examination, the U-Net also performed the best overall between the user's accuracy and producer's accuracy for the forest class especially. Interestingly, the forest class was more often confused with the developed class than the grass class, despite the grass class having more similar RGB values to the forest class than the developed class. Similar to the overall accuracies and kappa coefficient, the SVM and RF classes followed U-Net in terms of forest class accuracy, in that order. Figure 5 visualizes the differences between the three classifier results and the original NAIP image within Laurel boundaries. There are some inconsistencies within each classified image, but the U-Net classifier resulted in tree canopy without the well-documented salt and pepper effect, or large areas of misclassification.

3.2. Georgetown, TX Classification Experiment

The Georgetown, TX classification required considerably more time for training and classification due to its complex geographic extent. The U-Net CNN once again required the most time at 5 h 5 min and 7 s, the SVM classifier required 13 min and two seconds, and the RF classifier required 14 min and 1 s. We hypothesize that the U-Net CNN required significant more time because of the complexity of the image segmentation process needed to generate individual tiles and process each individually. Classification times, OA, and Kappa coefficient are reported in Table 6. We discovered that the U-Net classifier required a significantly longer time for classification, but only 9 min longer to train the classifier. The U-Net proved the time investment to be well worth it as the overall accuracy and kappa coefficients were well above the results reported for SVM and RF.

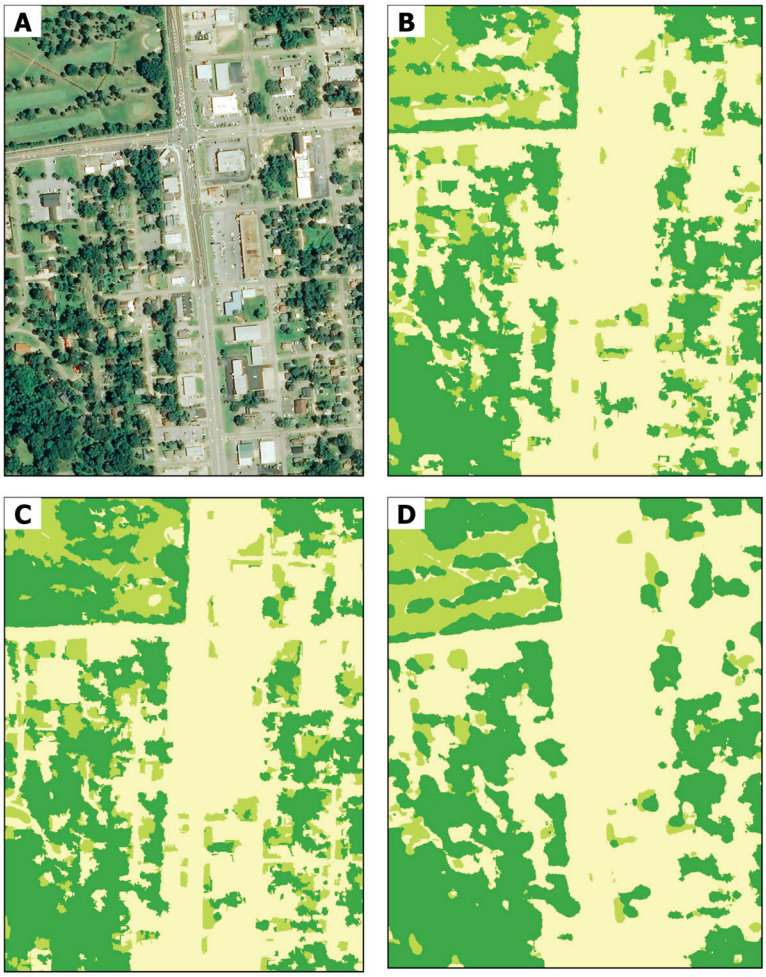


Figure 5. Comparison of the original NAIP image (**A**), the RF classifier results (**B**), SVM classifier results (**C**), and the U-Net classifier results (**D**) for Laurel, MS. Light green represents grass, dark green is urban tree canopy, and yellow is urban and other classes combined.

Table 6. Georgetown, TX classification accuracy assessment.

	U-Net	SVM	RF
Overall accuracy	89.8%	68.6%	71.3%
Kappa coefficient	0.8475	0.529	0.57
Classifier training time	13 m 53 s	4 m 16 s	3 m 48 s
Classification time	3 h 51 m 17 s	8 m 46 s	10 m 13 s

More specifically, each classifier performed differently for each class, as seen in Tables 7–9.

Table 7. U-Net classifier confusion matrix for Georgetown, TX.

	Grass	Developed	Forest	Total	User's Accuracy
Grass	335	0	7	342	98%
Developed	44	376	26	446	84%
Forest	21	24	367	412	89%
Total	400	400	400	1200	
Producer's accuracy	84%	94%	92%		

Despite their rapid training and classification, it is apparent that the SVM and RF classifiers struggled with the geographically complex Georgetown, TX imagery. The U-Net classifier likewise was not as successful in classifying Georgetown tree canopy as it was for

Laurel, MS, but it significantly outperformed the other two machine learning classifiers. Once again the developed class was most confusing for the forest class, but the U-Net classifier was able to navigate the difference between the two the best. This is visualized in Figure 6, where each classifier result is compared against the original NAIP image from 2012.

Table 8. SVM classifier confusion matrix for Georgetown, TX.

	Grass	Developed	Forest	Total	User's Accuracy
Grass	207	2	5	214	97%
Developed	168	368	147	683	54%
Forest	25	30	248	303	82%
Total	400	400	400	1200	
Producer's accuracy	52%	92%	62%		

Table 9. Random forest classifier confusion matrix for Georgetown, TX.

	Grass	Developed	Forest	Total	User's Accuracy
Grass	212	0	10	222	95%
Developed	182	385	131	698	55%
Forest	6	15	259	422	93%
Total	400	400	400	1200	
Producer's accuracy	53%	96%	65%		



Figure 6. Comparison of the original NAIP image (**A**), the RF classifier results (**B**), SVM classifier results (**C**), and the U-Net classifier results (**D**) for Georgetown, TX. Light green represents grass, dark green is urban tree canopy, and yellow is urban and other classes combined.

3.3. Case Study Results

Following the experiment results, we determined to use the U-Net CNN classifier for the 2022 and 2023 images for the study areas. The classifications were successful, and each required more time to complete to the classifications performed on the 2012 images (Table 10). However, they also resulted in similar accuracies for the forest classes, overall accuracies, and kappa coefficients. We suspect the longer processing times are related to the higher spatial resolution size (60 cm vs. 1 m) since the computer used for processing remained the same.

Table 10. Case study classification results.

	Time	User's Accuracy (Forest Class)	Producer's Accuracy (Forest Class)	OA	Kappa Coefficient
Laurel, MS	1 h 45 m 32 s	91%	95%	95%	0.92
Georgetown, TX	42 h 15 m 30 s	81%	92%	83%	0.75

After the post-classification change detection was performed, the canopy changes were analyzed with the urban heat surface maps (Figure 7). Georgetown, the growing southern city, showed extensive tree canopy changes from 2012 to 2022. Laurel, on the other hand, showed less significant changes. Many of the smaller UTC changes may be considered false positive changes. These occur where tree canopy was misclassified during either the 2012 or 2022/2023 classifications and therefore considered a change when there was simply a mistake in classification. Small changes were ignored in the qualitative assessment of tree canopy changes.

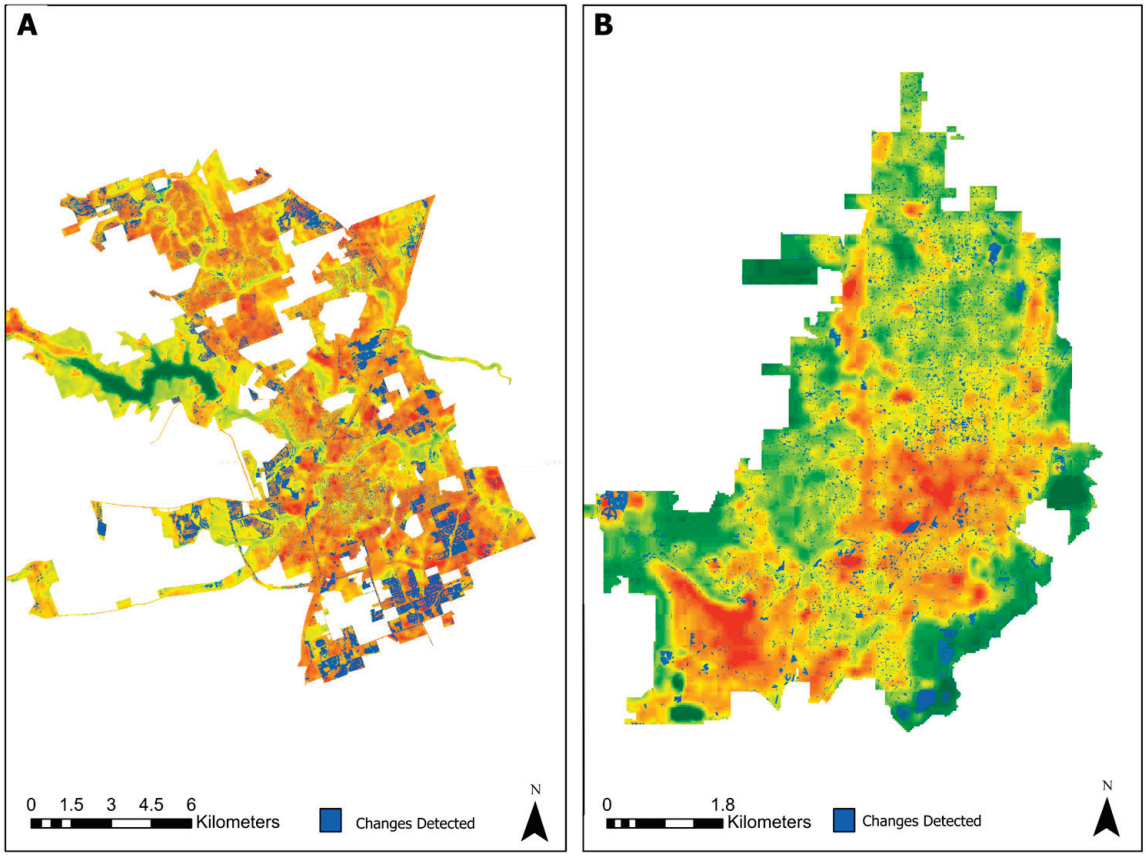


Figure 7. Canopy changes (in blue) overlaid on the heat maps for Georgetown (**A**) and Laurel (**B**).

Overall, Laurel experienced far less canopy change than Georgetown, but still experienced a small amount of loss connected with heat exposure. For example, Figure 8 shows the 2012 NAIP image (A), 2023 NAIP image (B), heat map (C), and heat map showing detected canopy loss (D) in a small area in the northwest portion of the city limits. The removal of trees for a small development of duplex homes may seem insignificant, but the higher heat has a strong spatial correlation with the loss of tree canopy in that region. The surrounding areas remain cool with the unchanging urban forest. Generally, Laurel has remained fairly stable with urban tree canopy area. Nevertheless, the difference between the temperatures in the small areas of canopy loss around the city and the rest of the city were statistically significant at the $p < 0.001$ level, indicating high confidence.

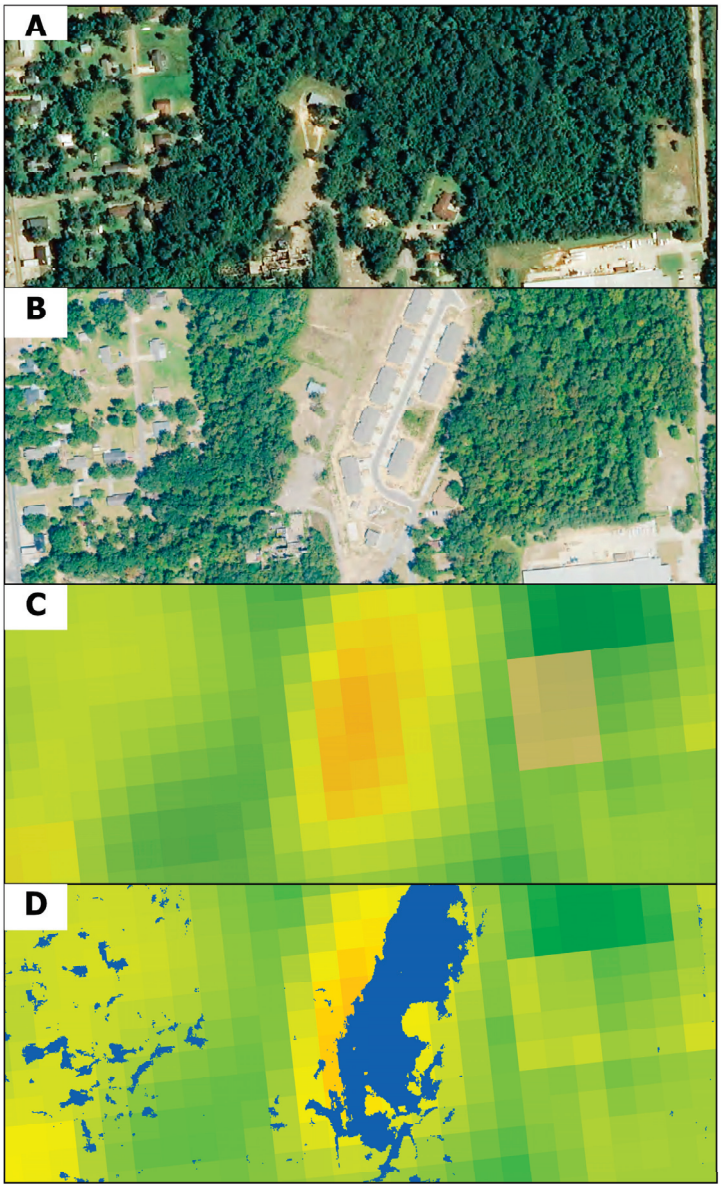


Figure 8. Laurel Mississippi example of tree canopy loss or change using the 2012 NAIP image (**A**), 2023 NAIP image (**B**), heat map (**C**), and heat map showing detected canopy loss (**D**).

Georgetown, TX was very different with the significant loss of canopy between the two data collection dates. Between that time, Georgetown grew significantly, and this is represented in the several new neighborhood developments. Although there were several examples of this, we represent just two in Figures 8 and 9. In Figure 9, we can see

heat especially where the newly developed subdivision or neighborhood was built, but it decreases in the area to the east where urban trees were still present. Just like Laurel, the difference in land surface temperatures in the tree canopy loss areas and the rest of the city limits was statistically significant. In fact, the tree canopy loss areas had temperatures that were higher and statistically significant at the $p < 0.001$ level.

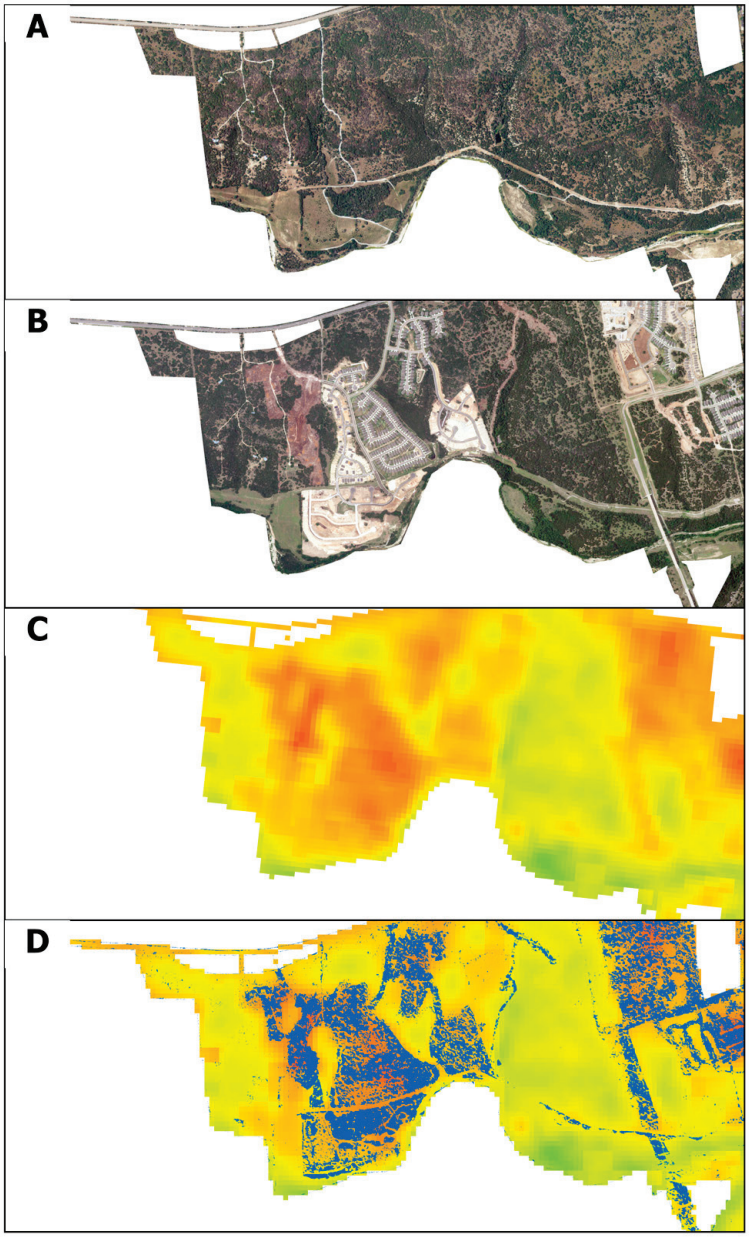


Figure 9. Georgetown, TX first example of tree canopy loss or change using the 2012 NAIP image (**A**), 2023 NAIP image (**B**), heat map (**C**), and heat map showing detected canopy loss (**D**).

Figure 10 displays a second example of this phenomenon in the northeastern region of the city where a small development had not been quite completed yet in 2012 but was finished by 2022. There are small pockets of urban forest that reduce heat to a small degree, but the new development overwhelms much of the good the urban forest is doing by making the urbanized region much warmer.

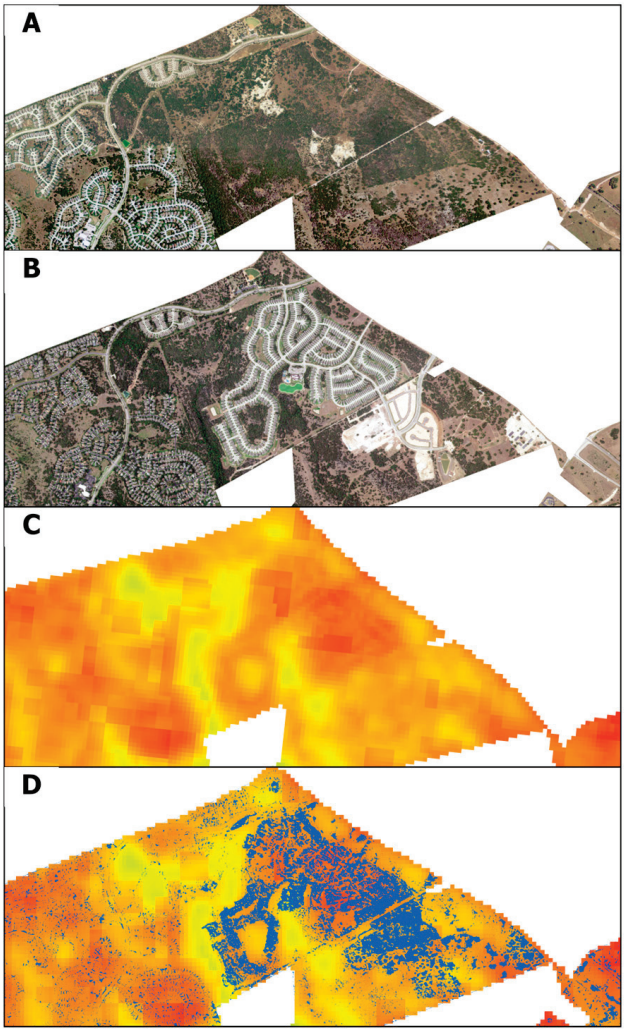


Figure 10. Georgetown, TX second example of tree canopy loss or change using the 2012 NAIP image (**A**), 2023 NAIP image (**B**), heat map (**C**), and heat map showing detected canopy loss (**D**).

4. Discussion

This study examined the effectiveness of two machine learning and one deep learning models adapted to remote sensing imagery classification for calculating UTC area within a small southern town and a quickly growing southern town in the United States of America. All three classifiers performed more quickly when processing the smaller town data from Laurel as the data size was smaller and more geographically compact. The overall accuracy for each classifier was highest for the small, more simple Laurel imagery, but for both Georgetown and Laurel, the U-Net classifier performed the best overall (89.8% and 91.4%, respectively), better than the SVM and RF classifiers. However, the SVM and RF classifiers were much faster than the U-Net classifiers in the ArcGIS Pro 3.1.3 setting. This may be caused by the computer hardware's limitations.

The findings from the experiment reflect well among similar experiments performed with other classifiers for determining UTC area coverage (Table 11). In their comparative study of different spatial scales for mapping accurate UTC using U-Net and object-based image analysis (OBIA), Wang et al. [35] found their U-Net model implemented with Python outperformed OBIA in every measure, reaching a result of 99% overall accuracy. Using both Google Earth imagery and LiDAR datasets, Timilsina et al. [38] obtained overall accuracies of 96% and 98% for 2005 and 2015, respectively. Most studies have relied solely on aerial or satellite imagery to detect and classify tree canopy and have resulted in high overall accuracies when processed with a deep learning classifier. The weakest results were

still impressive because of the intent to not only classify tree canopy but the species of tree [39]. Overall, the U-Net classification overall accuracy obtained in this study (89.8% and 91.4%) are comparable to other results presented in Table 11. The SVM and RF classifiers, however, performed poorly compared to the studies presented in Table 11. The SVM and RF classifiers repeatedly performed at about 70–80% overall accuracy, while the results shown below reach upwards of 94% [27,28,52]. This reveals some potential improvements to be made to the SVM and RF methods presented here. Because SVM is known to run well on fewer training samples, reducing the input for training this classifier may have improved the results. Nevertheless, the results of this study suggest that free-to-use NAIP RGB imagery is an accessible and comparable resource to use for obtaining UTC area coverage over small and growing cities. The NAIP imagery and robust tools available in ArcGIS Pro, especially the U-Net deep learning classifier, are able to provide accurate results for urban forestry managers without needing programming experience.

Table 11. Relevant studies using deep and machine learning to classify tree canopy area.

Data Sources and Study Areas	Imagery Dates	Methods	Highest Overall Accuracies Achieved	Authors
33 imagery patches of Vaihingen, Germany	2013	U-Net and OBIA	U-Net; 99%	[35]
Google Earth Satellite Imagery and LiDAR of Tasmania, Australia	2005; 2015	Object based CNN	2015 data; 98%	[38]
UAS LiDAR/Multispectral data over Toronto, Ontario, Canada	2020	Deep CNN, SVM, Maximum likelihood	Deep CNN, RGB and LiDAR; 97%	[23]
Gaofen-2 Satellite images over Beijing, China	2021	Improved Double Branch U-Net; other U-Net variants	Improved Double Branch U-Net; 95.8%	[25]
High-resolution satellite images for 888 cities across South America	2018–2020	Deeplab3+ deep learning	Deeplab3+; exceeds 90% for all 888 cities	[26]
NAIP over Columbia, SC, USA	2005; 2015	SVM	SVM; 94%	[52]
Planet SkySat satellite imagery over Chattanooga, TN, USA	2021–2022	SVM	SVM; 91%	[27]
Gaofen-2 satellite imagery over Hui'an county, Fujian Province, China	2021	Superpixel-enhanced deep neural forests (SDNF); RF	SDNF; 95%	[64]
Worldview-2 satellite imagery over Miami-Dade County, Florida, USA	2019; 2014	RF	RF; 87%	[28]
Worldview-3 satellite and LiDAR data over Brussels, Belgium	2015; 2017	Geospatial OBIA	GEOBIA; 71%	[39]
NAIP and SPOT satellite imagery over Wisconsin, USA	2013	RF, SVM, boosted regression trees	No significant differences between three classifiers	[37]

The case study presented UTC change results between the 2012 and 2022/2023 NAIP imagery sets for each city and compared the spatial distribution of UTC change to the urban heat maps derived from the Landsat 9 thermal bands. The patterns found in large amounts of growth in Georgetown, TX, and to a smaller extent in Laurel, were similar to the areas of growth in Columbia, SC in [52]. They are also consistent with findings by Tamaskani Esfehankalateh et al. [6] and Loughner et al. [5] that suggest a strong correlation between UTC and urban heat. Spatially, the city centers of both cities experienced the least change in UTC while the periphery experienced the most urban growth and UTC decline. This suggests a stable city, like Laurel, is likely to experience little change to the UHI effect within its borders as long as trees are protected like they are in Laurel. It also suggests that it is more difficult to mitigate UHI effect in the center of cities as there is less room for trees to be planted and grow once they are already removed for urban growth. Very little, if any, tree growth was reported in either city during the 10-year period. We also found that the differences between the temperatures in the areas tree canopy loss and the temperatures in

the areas of no change in both cities were statistically significant, indicating the role UTC plays in mitigating heat. The temperatures of the tree canopy loss areas were higher than those throughout the rest of the city. Future policies should prevent significant loss of tree canopy within city limits to mitigate further heat increases.

Several limitations were encountered during the course of this study. NAIP data are collected at times and dates that are beyond the control of any application. For example, despite being collected during leaf on conditions, the window for data collection is from march until October in most southern US cities, meaning that each year may be more or less easily classified depending on the tree conditions and greenness. Additionally, technological limitations with the computer hardware can impact data classification times. More modern computers, supercomputers, and cloud computing with ArcGIS can remove much of the frustration with processing time that we encountered in this study. Future work should focus on adding additional deep learning models to compare with U-Net to determine if another model may perform faster and provide a more accurate classification. Studies focusing on the identification and counting of individual urban trees have assessed multiple deep learning models, but it has yet to be investigated for mapping UTC area coverage [32–34,65–67].

5. Conclusions

This study tested three remote sensing image classifiers for mapping UTC area in both a small southern USA city and a midsized, growing USA city. This study determined that by using available tools from industry-leading ESRI ArcGIS Pro and free aerial imagery through the NAIP, overall accuracies similar to those obtained using expensive LiDAR and satellite data can be achieved. Of the three classifiers tested, the U-Net CNN classifier within ArcGIS Pro performed the best but required the longest processing time. The U-Net CNN overall accuracies of 89.8% and 91.4% compare favorably with other studies conducted with similar and more costly resources. Furthermore, insights into UTC changes and spatial correlations between urban heat and those UTC changes occurring in small and midsize cities in the southern US were obtained. For example, the peripheries of even midsized growing cities like Georgetown, TX are getting warmer as more UTC is removed for suburban neighborhoods. The broader impacts of this study reach the urban forestry managers at the local, state/province, and national levels as they seek to provide data-driven decisions for policy makers.

Author Contributions: Conceptualization: G.R.M.; Data curation: G.R.M., D.Z., L.N., C.S. and L.S.; Formal analysis: G.R.M., D.Z. and L.N.; Investigation: G.R.M.; Methodology: G.R.M.; Project administration: L.S.; Software: G.R.M., D.Z., L.N. and L.S.; Supervision: G.R.M.; Validation: G.R.M., D.Z. and L.N.; Visualization: G.R.M.; Writing—original draft: G.R.M., D.Z., L.N. and L.S.; Writing—review and editing: G.R.M., D.Z., L.N. and L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: All datasets used are publicly available.

Acknowledgments: A special thank you to all those who provided insightful feedback and comments that improved the value of the manuscript, including the reviewers.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Konijnendijk, C.C.; Ricard, R.M.; Kenney, A.; Randrup, T.B. Defining urban forestry—A comparative perspective of North America and Europe. *Urban For. Urban Green.* **2006**, *4*, 93–103. [CrossRef]
2. Nowak, D.J.; Greenfield, E.J. US urban forest statistics, values, and projections. *J. For.* **2018**, *116*, 164–177. [CrossRef]
3. Drillet, Z.; Fung, T.; Leong, R.; Sachidhanandam, U.; Edwards, P.; Richards, D. Urban vegetation types are not perceived equally in providing ecosystem services and disservices. *Sustainability* **2020**, *12*, 2076. [CrossRef]

4. Carne, R.J. Urban vegetation: Ecological and social value. In Proceedings of the National Greening Australia Conference, Fremantle, WA, Australia, 4–6 October 1994; pp. 4–6.
5. Loughner, C.P.; Allen, D.J.; Zhang, D.-L.; Pickering, K.E.; Dickerson, R.R.; Landry, L. Roles of urban tree canopy and buildings in urban heat island effects: Parameterization and preliminary results. *J. Appl. Meteorol. Climatol.* **2012**, *51*, 1775–1793. [CrossRef]
6. Tamaskani Esfehankalateh, A.; Ngarambe, J.; Yun, G.Y. Influence of tree canopy coverage and leaf area density on urban heat island mitigation. *Sustainability* **2021**, *13*, 7496. [CrossRef]
7. Janhäll, S. Review on urban vegetation and particle air pollution—Deposition and dispersion. *Atmos. Environ.* **2015**, *105*, 130–137. [CrossRef]
8. Lindén, J.; Gustafsson, M.; Uddling, J.; Watne, Å.; Pleijel, H. Air pollution removal through deposition on urban vegetation: The importance of vegetation characteristics. *Urban For. Urban Green.* **2023**, *81*, 127843. [CrossRef]
9. World Health Organization. Regional Office for Europe. *Urban Green Spaces and Health*; World Health Organization. Regional Office for Europe: Copenhagen, Denmark, 2016.
10. Wolf, K.L.; Lam, S.T.; McKeen, J.K.; Richardson, G.R.A.; van den Bosch, M.; Bardekjian, A.C. Urban trees and human health: A scoping review. *Int. J. Environ. Res. Public Health* **2020**, *17*, 4371. [CrossRef]
11. Othman, N.; Hamzah, H.; Mohd Salleh, M.Z. Relationship of trees as green infrastructure to pro-environmental behavior for psychological restoration in urbanized society: A systematic review. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *918*, 012047. [CrossRef]
12. Berland, A.; Shiflett, S.A.; Shuster, W.D.; Garmestani, A.S.; Goddard, H.C.; Herrmann, D.L.; Hopton, M.E. The role of trees in urban stormwater management. *Landsc. Urban Plan.* **2017**, *162*, 167–177. [CrossRef]
13. Carlyle-Moses, D.E.; Livesley, S.; Baptista, M.D.; Thom, J.; Szota, C. Urban trees as green infrastructure for stormwater mitigation and use. *For.-Water Interact.* **2020**, 397–432. [CrossRef]
14. McGee, J.A.; Day, S.D.; Wynne, R.H.; White, M.B. Using geospatial tools to assess the urban tree canopy: Decision support for local governments. *J. For.* **2012**, *110*, 275–286. [CrossRef]
15. Parmehr, E.G.; Amati, M.; Taylor, E.J.; Livesley, S.J. Estimation of urban tree canopy cover using random point sampling and remote sensing methods. *Urban For. Urban Green.* **2016**, *20*, 160–171. [CrossRef]
16. Klobucar, B.; Sang, N.; Randrup, T.B. Comparing ground and remotely sensed measurements of urban tree canopy in private residential property. *Trees For. People* **2021**, *5*, 100114. [CrossRef]
17. Li, X.; Chen, W.Y.; Sanesi, G.; Lafortezza, R. Remote Sensing in urban forestry: Recent applications and Future Directions. *Remote Sens.* **2019**, *11*, 1144. [CrossRef]
18. Moskal, L.M.; Styers, D.M.; Halabisky, M. Monitoring urban tree cover using object-based image analysis and public domain remotely sensed data. *Remote Sens.* **2011**, *3*, 2243–2262. [CrossRef]
19. Elmes, A.; Rogan, J.; Williams, C.; Ratick, S.; Nowak, D.; Martin, D. Effects of urban tree canopy loss on land surface temperature magnitude and timing. *ISPRS J. Photogramm. Remote Sens.* **2017**, *128*, 338–353. [CrossRef]
20. Guo, L.; Chehata, N.; Mallet, C.; Boukir, S. Relevance of airborne lidar and multispectral image data for urban scene classification using random forests. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 56–66. [CrossRef]
21. Chuang, W.-C.; Boone, C.G.; Locke, D.H.; Grove, J.M.; Whitmer, A.; Buckley, G.; Zhang, S. Tree Canopy Change and neighborhood stability: A comparative analysis of Washington, D.C. and Baltimore, MD. *Urban For. Urban Green.* **2017**, *27*, 363–372. [CrossRef]
22. Ghanbari Parmehr, E.; Amati, M. Individual tree canopy parameters estimation using UAV-based photogrammetric and Lidar Point Clouds in an urban park. *Remote Sens.* **2021**, *13*, 2062. [CrossRef]
23. Elamin, A.; El-Rabbany, A. UAV-based multi-sensor data fusion for urban land cover mapping using a deep convolutional neural network. *Remote Sens.* **2022**, *14*, 4298. [CrossRef]
24. Hartling, S.; Sagan, V.; Maimaitijiang, M. Urban tree species classification using UAV-based multi-sensor data fusion and machine learning. *GIScience Remote Sens.* **2021**, *58*, 1250–1275. [CrossRef]
25. Chen, S.; Chen, M.; Zhao, B.; Mao, T.; Wu, J.; Bao, W. Urban tree canopy mapping based on double-branch convolutional neural network and multi-temporal high spatial resolution satellite imagery. *Remote Sens.* **2023**, *15*, 765. [CrossRef]
26. Guo, J.; Hong, D.; Liu, Z.; Zhu, X.X. Continent-wide urban tree canopy fine-scale mapping and coverage assessment in south America with high-resolution satellite images. *ISPRS J. Photogramm. Remote Sens.* **2024**, *212*, 251–273. [CrossRef]
27. Mix, C.; Hunt, N.; Stuart, W.; Hossain, A.K.M.A.; Bishop, B.W. A spatial analysis of urban tree canopy using high-resolution land cover data for Chattanooga, Tennessee. *Appl. Sci.* **2024**, *14*, 4861. [CrossRef]
28. Hochmair, H.H.; Benjamin, A.; Gann, D.; Juhasz, L.; Olivas, P.; Fu, Z.J. Change analysis of urban tree canopy in Miami Dade county. *Forests* **2022**, *13*, 949. [CrossRef]
29. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.-S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
30. Zhang, L.; Zhang, L.; Du, B. Deep Learning for Remote Sensing Data: A technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40. [CrossRef]
31. Youssef, R.; Aniss, M.; Jamal, C. Machine learning and deep learning in remote sensing and urban application. In Proceedings of the 4th Edition of International Conference on Geo-IT and Water Resources 2020, Geo-IT and Water Resources 2020, Al-Hoceima, Morocco, 11–12 March 2020. [CrossRef]
32. Yan, S.; Jing, L.; Wang, H. A new individual tree species recognition method based on a convolutional neural network and high-spatial resolution remote sensing imagery. *Remote Sens.* **2021**, *13*, 479. [CrossRef]

33. Lv, L.; Li, X.; Mao, F.; Zhou, L.; Xuan, J.; Zhao, Y.; Yu, J.; Song, M.; Huang, L.; Du, H. A deep learning network for individual tree segmentation in UAV images with a coupled CSPNet and attention mechanism. *Remote Sens.* **2023**, *15*, 4420. [CrossRef]
34. Zamboni, P.; Junior, J.M.; Silva, J.d.A.; Miyoshi, G.T.; Matsubara, E.T.; Nogueira, K.; Goncalves, W.N. Benchmarking anchor-based and anchor-free state-of-the-art deep learning methods for individual tree detection in RGB high-resolution images. *Remote Sens.* **2021**, *13*, 2482. [CrossRef]
35. Wang, Z.; Fan, C.; Xian, M. Application and evaluation of a deep learning architecture to urban tree canopy mapping. *Remote Sens.* **2021**, *13*, 1749. [CrossRef]
36. Morgan, G.R.; Wang, C.; Li, Z.; Schill, S.R.; Morgan, D.R. Deep learning of high-resolution aerial imagery for coastal Marsh Change Detection: A comparative study. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 100. [CrossRef]
37. Erker, T.; Wang, L.; Lorentz, L.; Stoltman, A.; Townsend, P.A. A statewide urban tree canopy mapping method. *Remote Sens. Environ.* **2019**, *229*, 148–158. [CrossRef]
38. Timilsina, S.; Aryal, J.; Kirkpatrick, J.B. Mapping urban tree cover changes using object-based convolution neural network (OB-CNN). *Remote Sens.* **2020**, *12*, 3017. [CrossRef]
39. Choudhury, M.A.M.; Marcheggiani, E.; Galli, A.; Modica, G.; Somers, B. Mapping the urban atmospheric carbon stock by LiDAR and WorldView-3 data. *Forests* **2021**, *12*, 692. [CrossRef]
40. Kafy, A.; Rahman, M.S.; Faisal, A.-A.-; Hasan, M.M.; Islam, M. Modelling future land use land cover changes and their impacts on land surface temperatures in Rajshahi, Bangladesh. *Remote Sens. Appl. Soc. Environ.* **2020**, *18*, 100314. [CrossRef]
41. Alexander, C. Normalised difference spectral indices and urban land cover as indicators of land surface temperature (LST). *Int. J. Appl. Earth Obs. Geoinf.* **2020**, *86*, 102013. [CrossRef]
42. Dutta, D.; Rahman, A.; Paul, S.K.; Kundu, A. Impervious surface growth and its inter-relationship with vegetation cover and land surface temperature in peri-urban areas of Delhi. *Urban Clim.* **2021**, *37*, 100799. [CrossRef]
43. Mukherjee, F.; Singh, D. Assessing land use–land cover change and its impact on land surface temperature using LANDSAT DATA: A comparison of two urban areas in India. *Earth Syst. Environ.* **2020**, *4*, 385–407. [CrossRef]
44. Imran, H.M.; Hossain, A.; Islam, A.K.; Rahman, A.; Bhuiyan, M.A.; Paul, S.; Alam, A. Impact of land cover changes on land surface temperature and human thermal comfort in Dhaka City of Bangladesh. *Earth Syst. Environ.* **2021**, *5*, 667–693. [CrossRef]
45. Britannica, T. Editors of Encyclopaedia. Laurel. In *Encyclopedia Britannica*; Encyclopædia Britannica, Inc.: London, UK, 2011; Available online: https://www.britannica.com/place/Laurel-Mississippi (accessed on 1 April 2024).
46. Climate-data.org. (n.d.). Available online: https://en.climate-data.org/north-america/united-states-of-america/mississippi/laurel-17322/ (accessed on 1 April 2024).
47. Home Town. 2024. Available online: https://www.hgtv.com/shows/home-town (accessed on 1 April 2024).
48. Mississippi Forestry Commission. MFC Recognizes City of Laurel for Tree City USA® Participation. 2019. Available online: https://www.mfc.ms.gov/2019/09/laurel-tree-city-usa/ (accessed on 1 April 2024).
49. US Census Bureau. Large Southern Cities Lead Nation in Population Growth. In *Census.Gov*; 18 May 2023. Available online: www.census.gov/newsroom/press-releases/2023/subcounty-metro-micro-estimates.html (accessed on 1 April 2024).
50. City of Georgetown Texas. 2024. Available online: https://planning.georgetown.org/tree-removal-pruning-and-landscape/ (accessed on 1 April 2024).
51. Davis, D. National Agriculture Imagery Program (NAIP) Information Sheet. Available online: https://www.fsa.usda.gov/Assets/USDA-FSA-Public/usdafiles/APFO/support-documents/pdfs/naip_infosheet_2016.pdf (accessed on 3 April 2024).
52. Morgan, G.R.; Fulham, A.; Farmer, T.G. Machine learning in urban tree canopy mapping: A Columbia, SC case study for urban heat island analysis. *Geographies* **2023**, *3*, 359–374. [CrossRef]
53. ESRI. (n.d.-a) How U-net works? Available online: https://developers.arcgis.com/python/guide/how-unet-works/ (accessed on 22 September 2024).
54. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597.
55. Moradi, F.; Javan, F.D.; Samadzadegan, F. Potential evaluation of visible-thermal UAV image fusion for individual tree detection based on convolutional neural network. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *113*, 103011. [CrossRef]
56. Clark, A.; Phinn, S.; Scarth, P. Optimized U-net for land Use–Land cover classification using aerial photography. *PFG–J. Photogramm. Remote Sens. Geoinf. Sci.* **2023**, *91*, 125–147. [CrossRef]
57. Wang, X.; Hu, Z.; Shi, S.; Hou, M.; Xu, L.; Zhang, X. A deep learning method for optimizing semantic segmentation accuracy of remote sensing images based on improved unet. *Sci. Rep.* **2023**, *13*, 7600. [CrossRef]
58. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning: With Applications in R.*; Springer: New York, NY, USA, 2013.
59. ESRI. (n.d.-b) Train Support Vector Machine Classifier (Spatial Analyst). Available online: https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/train-support-vector-machine-classifier.htm (accessed on 22 September 2024).
60. Jensen, J.R. *Introductory Digital Image Processing: A Remote Sensing Perspective*; Pearson Education: Glenview, IL, USA, 2016.
61. Congalton, R.; Oderwald, R.G.; Mead, R. Assessing Landsat classification accuracy using discrete multivariate statistical techniques. *Photogramm. Eng. Remote Sens.* **1983**, *49*, 1671–1678.
62. Peiman, R. Pre-classification and post-classification change-detection techniques to monitor land-cover and land-use change using multi-temporal landsat imagery: A case study on pisa province in Italy. *Int. J. Remote Sens.* **2011**, *32*, 4365–4381. [CrossRef]

63. Serra, P.; Pons, X.; Saurí, D. Post-classification change detection with data from different sensors: Some accuracy considerations. *Int. J. Remote Sens.* **2003**, *24*, 3311–3340. [CrossRef]
64. Liu, Y.; Zhang, H.; Cui, Z.; Lei, K.; Zuo, Y.; Wang, J.; Hu, X.; Qiu, H. Very high-resolution images and superpixel-enhanced deepneural forest promote urban tree canopy detection. *Remote Sens.* **2023**, *15*, 519. [CrossRef]
65. Ventura, J.; Pawlak, C.; Honsberger, M.; Gonsalves, C.; Rice, J.; Love, N.L.R.; Han, S.; Nguyen, V.; Sugano, K.; Doremus, J.; et al. Individual tree detection in large-scale urban environments using high-resolution multispectral imagery. *Int. J. Appl. Earth Obs. Geoinf.* **2024**, *130*, 103848. [CrossRef]
66. Wallace, L.; Sun, Q.; Hally, B.; Hillman, S.; Both, A.; Hurley, J.; Martin Saldias, D.S. Linking urban tree inventories to remote sensing data for individual tree mapping. *Urban For. Urban Green.* **2021**, *61*, 127106. [CrossRef]
67. Yang, M.; Mou, Y.; Liu, S.; Meng, Y.; Liu, Z.; Li, P.; Xiang, W.; Zhou, X.; Peng, C. Detecting and mapping tree crowns based on convolutional neural network and google earth images. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *108*, 102764. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of Geo-Information

MDPI

Article

Investigating the Performance of Open-Vocabulary Classification Algorithms for Pathway and Surface Material Detection in Urban Environments

Kauê de Moraes Vestena [1,*], Silvana Phillipi Camboim [1], Maria Antonia Brovelli [2] and Daniel Rodrigues dos Santos [3]

[1] Department of Geomatics, Federal University of Paraná, Curitiba 80060-000, Brazil; silvanacamboim@ufpr.br
[2] Politecnico di Milano, Department of Civil and Environmental Engineering, 20133 Milan, Italy; maria.brovelli@polimi.it
[3] Military Institute of Engineering, Cartography Section, Rio de Janeiro 22290-270, Brazil; daniel.rodrigues@ime.eb.br
* Correspondence: kauemv2@gmail.com

Abstract: Mapping pavement types, especially in sidewalks, is essential for urban planning and mobility studies. Identifying pavement materials is a key factor in assessing mobility, such as walkability and wheelchair usability. However, satellite imagery in this scenario is limited, and in situ mapping can be costly. A promising solution is to extract such geospatial features from street-level imagery. This study explores using open-vocabulary classification algorithms to segment and identify pavement types and surface materials in this scenario. Our approach uses large language models (LLMs) to improve the accuracy of classifying different pavement types. The methodology involves two experiments: the first uses free prompting with random street-view images, employing Grounding Dino and SAM algorithms to assess performance across categories. The second experiment evaluates standardized pavement classification using the Deep Pavements dataset and a fine-tuned CLIP algorithm optimized for detecting OSM-compliant pavement categories. The study presents open resources, such as the Deep Pavements dataset and a fine-tuned CLIP-based model, demonstrating a significant improvement in the true positive rate (TPR) from 56.04% to 93.5%. Our findings highlight both the potential and limitations of current open-vocabulary algorithms and emphasize the importance of diverse training datasets. This study advances urban feature mapping by offering a more intuitive and accurate approach to geospatial data extraction, enhancing urban accessibility and mobility mapping.

Keywords: open-vocabulary algorithms; pavement segmentation; surface material detection; street-view imagery; urban mobility

1. Introduction and Related Work

The study of sidewalks is crucial for accessibility and quality of life in cities [1–4]. As the required level of detail is not available in satellite imagery and in situ surveys are costly, mapping such features is challenging. Using street-level imagery for this purpose is, therefore, a potential improvement. However, defining the features to be classified in street-level imagery is challenging because landscape features are conceptualized in natural language. Thus, large language models (LLMs) can be used to improve the process of image pattern retrieval from street-level imagery by allowing the use of flexible and natural language prompts. This characteristic makes the extraction of map features more intuitive and accurate by exploiting the language-understanding capabilities of LLMs, which can be applied in many contexts, such as in urban planning and transport studies, such as in [5], where it was used to solve a community-level land-use task through using a feedback iteration. Recently, there has been considerable interest in the accessibility of these models,

as demonstrated by ChatGPT, which serves as an interface for user interaction with the latest iterations of the Generative Pre-trained Transformer (GPT). This development is the most recent in a continuing series of advances in natural language processing. The study of natural language processing (NLP) can be traced back to the 1950s [6], with early developments culminating in the establishment of the essential subtasks, such as sentence boundary detection, tokenization, part-of-speech assignment to individual words, morphological decomposition, chunking, and problem-specific segmentation [6]. Later developments resulted in the tailoring of conventional algorithms, such as support vector machines and hidden Markov models.

Among these developments, the extension of open-vocabulary prompting for image pattern retrieval represents a substantial leap forward in NLP, driven by advances in LLMs. This approach has recently gained more attention due to its capabilities of richer expressiveness and more extensive flexibility due to its generalization capabilities [7], its ability to capture nuances of processes [8], and its enabling of transfer learning [9]. The work of Zareian et al. [10] claims to be the first to pose the problem of "Open Vocabulary Object Detection" in opposition to other similar approaches. One example of that alternative approach was "Zero Shot-Detection", which aims to generalize from seen to unseen classes using semantic descriptions or attributes [11–13]. Another approach would be weakly supervised detection, which focuses on detecting classes with limited information, such as image-level or noisy labels [14,15]. It is worth noticing that there are some main drawbacks to open-vocabulary approaches, such as a higher computational cost due to their larger complexity [7] and the risk of misinterpretation due to the lack of constraints [8].

This paper aims to evaluate open-vocabulary algorithms within a domain-specific pipeline for pathway segmentation and surface material classification. To achieve these objectives, we designed a two-stage pipeline that establishes a workflow for large-scale, georeferenced pathway surface classification. In Experiment 1, we segment pathways in street-level imagery, while in Experiment 2, we classify segmented pathways by surface material type. Performance is evaluated regarding the pipeline's accuracy, adaptability to various pathway types, and efficiency in processing large-scale georeferenced images, aiming to develop a robust workflow for urban infrastructure mapping. Identification of pavement material types is crucial for maintaining road safety and ensuring the well-being of people [16]. This process is vital for mobility studies as it influences safety, skid resistance, and road noise [17].

Particularly for sidewalks, identifying suitable pavement types is essential for improving accessibility, urban mobility, and safety for all users, including those with reduced mobility [18]. Additionally, using specific pavement types, like exposed aggregate concrete (EAC) and porous concrete, can significantly reduce noise and enhance safety on both roadways and sidewalks [19]. Zeng & Boehm [20] exploited a broader investigation of open-vocabulary algorithms, which obtained averagely good results compared to SOTA closed-vocabulary ones, observing a prompt-dependent accuracy.

Although "street" and "sidewalk" are common classes amongst many classification/segmentation datasets [21–24], being one of the reference classes for testing many proposed algorithms [24–26], pavement type identification is a far less commonly undertaken task: convolutional neural networks were used to identify a few classes, limited to asphalt, gravel, and cement [27], and also for the identification of asphalt damage traits, such as "pothole" or "patch" [28], and pavement damage assessment [29], as well as for pixel-level segmentation, limited to "paved" and "unpaved" streets [30], and have only recently included eight specific classes, such as "granite blocks" and "hexagonal asphalt paver" [31]. Each of these studies developed specific models with a closed set of classes, thus not allowing a wide use of the algorithms, in contrast to open-vocabulary counterparts, such as CLIP [32] and Grounding Dino [33], which are designed to be generalist algorithms.

Furthermore, none of these previous studies on pavement material identification follow a global standard. This work innovates by integrating its application into OpenStreetMap (OSM), the world's most popular open and collaborative geographic database [34].

The feature attribute standards created by the OSM community have typically been agreed upon over many years, including surface materials, although they are often not defined by domain experts. Adopting this strategy helps maintain interoperability between the developments of this research and other tools and applications linked to OSM. In this context, this work aims to test the behaviour of some state-of-the-art algorithms for open-vocabulary image classification tasks in identifying pavement types according to the OSM conceptual model, thus aiming to help design valuable building blocks for producing compliant geographic data.

This study presents a valuable proposition to the community of urban planners and geospatial analysts. By using open-vocabulary algorithms and large language models (LLMs) to segment and identify pavement materials in street-level imagery, this work advances the methodology for mapping urban features. It bridges the gap between natural language processing and geospatial data extraction, providing a more intuitive and granular approach to feature classification. By integrating with OpenStreetMap (OSM), the study improves data interoperability and sets a precedent for using open and collaborative platforms for urban data management. We also contribute to creating building blocks for producing compliant geodata by providing open resources, such as the Deep Pavements dataset and a fine-tuned CLIP-based model. These innovations contribute to more accessible, scalable, and flexible solutions for urban infrastructure analysis, with potential applications in transport planning, accessibility improvement, and public safety.

2. Methodology

The task of using NLP to extract features from images has as its main challenge the process called "grounding", whose level means the capability of the model to associate elements of language with the proper regions in the images [35–37]. It was shown that the relevance of the hierarchical relationships between concepts improves the grounding level by establishing ontologies [38–40]. Ontologies bridge the gap between natural language and logical reasoning by providing a machine-readable representation (class) of real-world concepts [41]. Our study's methodological approach is presented in Figure 1, which includes the main steps and processes to fulfil our study's goals.

The motivational task behind this study can be formalized as follows: "Given a set of street view images, the objective is to segment all visible paths in each image using free-text input algorithms, identify their surface materials, and ensure that the process is not affected by the use of synonyms in the input". Considering the hierarchical relationships between the entities of interest and their properties, an ontology was created using the Web Ontology Language (OWL) standard [42]. This ontology is presented as a graph in Figure 2. The hierarchy naturally starts with "pathway", which is then specialized into "Road" and "Footway" categories. The branching follows up to two main interest categories: "Road" and "Sidewalk". All of them, at the same time, have their main differences. Still, they fundamentally share the characteristic of being walkable, which is essentially made possible through having a surface material, hereby treated as a fundamental shared characteristic of all "pathways". After that, "surface material" has its branching sub-ontology that is composed of two main categories, namely "unpaved material" and "paved material", the latter being subdivided into "continuous" and "block-based". This property sharing is expected to be visible in the results of any algorithm classifying bitmaps among these categories. With the ontology set, it is possible to establish different degrees of semantic separation among classes, highlighting pavement surfaces. Therefore, "asphalt" is more similar to "concrete" (0 degrees, same hierarchical level) than it is to "sett" (1 degree) and compacted pavement (2 degrees). This ontology is basilar to the proposed experiment, with its hierarchical relationships being called off along the study. As far as our knowledge goes, no similar ontology has been proposed before.



Figure 1. The study methodology workflow.

To undertake the present study, we designed two sequential experiments as part of a comprehensive methodology to obtain surface-labeled, segmented pathways from georeferenced terrestrial imagery. The first experiment, "Free Prompting and Random Imagery Evaluation", focuses on pathway segmentation. Here, we aimed to test the feasibility of open-vocabulary prompting across a diverse set of 296 randomly selected images, covering multiple categories and prompts to observe general trends and the model's responses in a realistic, large-scale processing scenario. In cases where an image was excessively blurry, it was discarded from the sample to ensure reliable segmentation results. The employed algorithm was a combination of Grounding Dino [33] and the foundation model SAM—Segment Anything [43]. Grounding Dino provided output bounding boxes based on the prompts, which were then used as inputs for SAM to obtain the segmented features. We used Mappillary, an ever-growing platform with billions of community-contributed CC BY-SA licensed street-view images [44], as the data source.

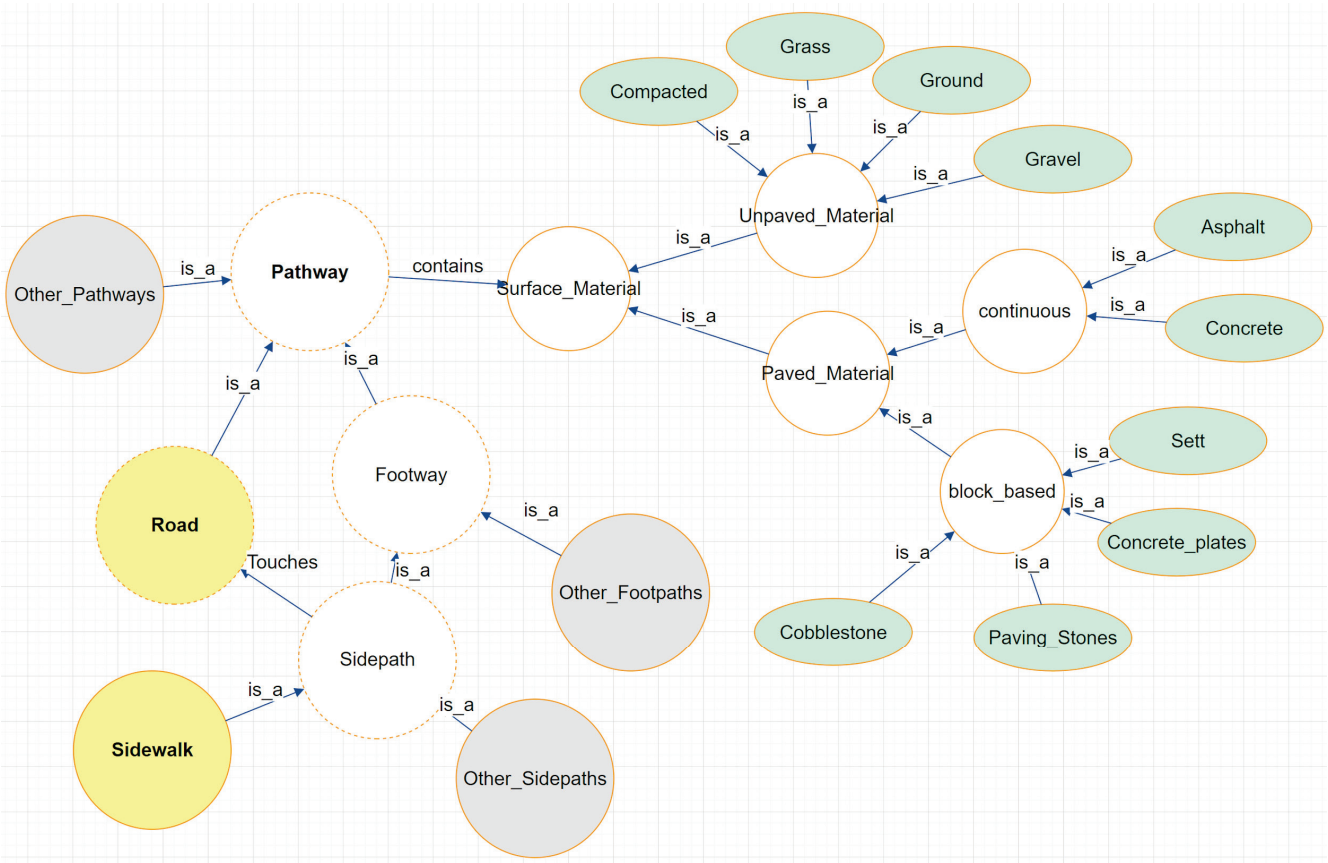


Figure 2. The ontology of the phenomena of interest.

The second experiment, designed for surface-labeling, namely "Evaluation of Open-Vocabulary Algorithms for Standardized Pavement Classification", builds upon the segmentation results of the first experiment. Here, we tested standardized pavement classes prompting against images of surface patches to assess the classification ability of an open-vocabulary algorithm in a constrained setup. We employed the CLIP algorithm, known for assigning probabilities to each entry of a set of sentences that can be changed at each query, constituting a fundamental advantage, as it allows adaptation to different realities. We tested eight variations of the algorithm, each employing different datasets and training strategies, as compared in Table 1 regarding their origin and computational costs. The data source for this experiment was the "deep-pavements-dataset", a tailored dataset of surface patches collaboratively maintained on GitHub [45], containing 500 samples for OSM-compliant pavement categories at the time of the experiments. The testing included all categories in the ontology, with examples shown in Figure 3. These experiments establish a pipeline where Experiment 1 segments pathways and Experiment 2 assigns surface labels, demonstrating a workflow for large-scale pathway surface classification using georeferenced images.

Table 1. Tested CLIP algorithm variants.

Package	Model Name	Base Algorithm	Main Image Dataset	Model Size (GB)	Image Parameters (Millions)	Text Parameters (Millions)	Image GFLOPS	Text GFLOPS	Package	Model Name
CLIP	RN101	CNN	OpenAI	0.28	56.26	63.43	19.54	5.96	CLIP	RN101
CLIP	RN50x64	CNN	OpenAI	1.26	420.38	202.88	529.11	23.55	CLIP	RN50x64
CLIP	ViT-B/32	VT	OpenAI	0.34	87.85	63.43	8.82	5.96	CLIP	ViT-B/32
CLIP	ViT-L/14	VT	OpenAI	0.89	303.97	123.65	162.03	13.3	CLIP	ViT-L/14
Open CLIP	ViT-H-14-378-quickgelu	VT	dfn5b	3.95	632.68	354.03	1006.96	47.09	Open CLIP	ViT-H-14-378-quickgelu
Open CLIP	EVA02-E-14-plus	VT	laion2b_s9b_b144k	10.1	4350.56	694.33	2264.33	97.86	Open CLIP	EVA02-E-14-plus



Figure 3. Some snapshots of the Deep Pavements dataset.

It is valuable to elaborate on the reasons for selecting this set of algorithms for our study. SAM is particularly noted for its unpaired segmentation capabilities, supported by a billion-level sample size, which enhances its applicability to real-world scenarios and provides superior generalization performance [46,47]. Grounding Dino offers complementary strengths and significantly outperforms comparable algorithms. This advantage is attributed to its use of a transformer architecture that integrates multi-level text information [33,48]. CLIP marked a significant breakthrough in the vision-language domain by employing a shared embedding space for text and images created through a contrastive learning approach [49]. It has been recognised for its robustness across various scenarios [50–52]. These algorithms are considered zero-shot learners capable of performing tasks without being specifically optimized [11]. Furthermore, despite their relatively recent introduction, these models have already seen widespread use in the industry for various applications, such as in automated image data annotation, image search engines, accessibility tools providing image descriptions, and enhancing content recommendation systems [53].

Regarding the first experiment and the proposed ontology, we tested the following categories with their corresponding prompts:

1. Auxiliary: These entities are detected to determine if a pavement detection failure occurred due to occlusion rather than incorrect detection. Prompts are “car”, “vehicle”, “pole”, and “tree”.
2. Sidepaths: This query is primarily aimed at detecting sidewalks, but it may also retrieve other sidepaths, such as paved shoulders. Prompts are “sidewalk”, “sidepath”, “sideway”, “sidetrack”, and “lateral”.
3. Roads: Focused on detecting motorized pathways. Prompts are “road” and “street”.
4. Pathways: These are intended to detect any kind of traversable way. Prompts are “way”, “path”, “pathway”, “pavement”, and “track”.
5. Surface pavement types: Prompts directly target the property, with additional ones included for broader testing. Prompts are “sett”, “grass”, “cobblestone”, “earth”, “soil”, “dirt”, “sand”, “concrete”, “paving stones”, “chip seal”, “gravel”, “compacted”, “asphalt”, “concrete plates”, and “ground”.
6. Abstract concepts: Words that do not have a unique or physical representation are used to test the model’s responses. Prompts include anything, nothing, something, void, and thing.

For the second experiment, we conducted two tests with different strategies to compare the classification accuracy with the pre-trained models. In the first “Model Aggregation” test, we summed the class probabilities across all models to create an overall score. The class

with the highest aggregated score was selected as the final prediction. This aggregation was performed internally, using all models initially before being repeated with the three best overall performers.

In the second approach, termed the "Fine-Tuning" test, we selected 60% of the samples for fine-tuning and used the remaining 200 samples per category for testing. Following the specifications in Table 1, we chose the lightest model to evaluate the extent of improvement relative to the model size and computational burden. Previous studies [49] showed that CLIP's fine-tuning is highly sensitive to the optimizer's algorithm hyperparameters. Therefore, we empirically tested multiple scenarios, presenting the worst and the best outcomes side by side. All the produced models and analyses are published in a repository at the HuggingFace community [54] to ensure reproducibility.

3. Results and Discussion

3.1. Free Prompting and Random Imagery Evaluation

Following the methodology presented in the previous section, eight images were tested for each prompt. As such, for categories that received 5 prompts, the experiment involved 40 images. Table 2 presents the results obtained for the "auxiliary" category. In the forthcoming tables, a "**" sign indicates that the targeted prompt does not exist in the image sample; thus, the correct outcome should be none.

Table 2. Prompt results for the "auxiliary" category.

Row	Prompt	1	2	3	4	5	6	7	8	Hit %
I	car	car	truck	car	car	car panel	car	car	car	100
II	poles	street lamp	power pole	building pole	street lamp	street lamp	guard rail **	street lamp	street lamp	87.5
III	vehicle	car	truck	car	truck	fence **	car	car	truck	87.5
IV	tree	tree	Tree	tree	tree	tree	vegetation	tree	tree	100

**: it was't present on the sample, thus the proper answer would be "none".

The desired results were achieved in all cases, except when the targeted feature was absent in the picture, causing the algorithm to hallucinate. Table 3 presents the results for the "Sidepaths" category.

Table 3. Prompt results for the "sidepaths" category.

Row	Prompt	1	2	3	4	5	6	7	8	Hit %
I	sidewalk	sidewalk	sidewalk	road	sidewalk	sidewalk	a sidepath	kerbs	sidewalk	75
II	sidepath	road **	road **	kerb	road **	road	road **	road	road	0
III	sideway	car panel	car	truck	car	car	car panel	road	car	0
IV	sidetrack	car	car panel	building	car	car panel **	truck container	car	car	0
V	lateral	road **	road	road and sidewalk	car **	car	vegetation	road	car **	0
VI	walkway	road and sidepath	road **	road **	road **	partial sidewalk	all pavements	road	road **	12.5

**: it was't present on the sample, thus the proper answer would be "none".

There is considerable evidence supporting the hypothesis of a lack of generalization capability. While "sidewalk" produced consistently good results, none of the other prompts were recognised as similar concepts, which should have led to proper detections. Recall that "sidewalk" is common in many well-spread training datasets. Table 4 presents the results obtained for the "roads" category, reinforcing the hypothesis of a stronger capability for road identification, likely due to the prevalence of roads in terrestrial imagery.

Table 4. Prompt results for the "sidepaths" category.

Row	Prompt	1	2	3	4	5	6	7	8	Hit %
I	street	road	road	road	road	road	road	road	road	100
II	road	road	road	road	road	road	road	road	road	100

Table 5 presents the results for the "pathway" category. Although the results were generally good, there was a noticeable bias towards detecting roads.

Table 5. Prompt results for the "pathway" category.

Row	Prompt	1	2	3	4	5	6	7	8	hit %
I	track	car	road	road	road	road	road	road	road	87.5
II	pavement	road	road	a path	road	road --	road	walkway	road	100
III	way	bus	road markings	road	road	bus	road	sidewalk	road	62.5
IV	path	road	road	road	road	road	road	road	road	100
V	pathway	road	road	sidewalk	road	road	road	road	road	100

Although, on average, we can see the expected results, there is a clear bias towards detecting roads. Table 6 presents the results for the "surface" category, where only "asphalt" and "grass" yielded satisfactory results. Other prompts, such as "sand" and "concrete", were less successful due to the nature of the test, which involved random images that often did not correspond to the prompts. The correct behaviour of returning "nothing" was only observed in the "concrete plates" category.

Table 6. Prompt results for the "surface" category.

Row	Prompt	1	2	3	4	5	6	7	8	Hit %
I	sett	car **	car **	car **	car **	asphalt road **	tree **	car **	car **	0
II	grass	grass sidepath	grass sidewalk	grass sidewalk	grass in a garden	grass sidepath	plants **	grass stripe	grass sidepath	87.5
III	cobblestone	grass sidepath	a pathway with stones	bush row	dirt sidepath	sidepath with vegetation	asphalt road	asphalt road	asphalt road	0
IV	earth	car **	gravel sidewalk **	asphalt road **	car panel **	asphalt road **	a tire **	asphalt road **	car **	0
V	soil	an unpaved sidepath	asphalt road **	asphalt road	soy crop	car	soil sidepath	grass sidewalk	grass sidepath	12.5
VI	dirt	asphalt road **	asphalt road **	car **	unpaved sidewalk **	asphalt road **	asphalt road and car panel **	a grass slope **	asphalt road **	0
VII	sand	car **	asphalt road **	asphalt road **	sand sidewalk	asphalt road **	asphalt road **	asphalt road (partial) **	car	12.5
VIII	concrete	asphalt road **	compacted pathway **	asphalt road **	asphalt road **	concrete sidewalk	guard rail **	asphalt road **	asphalt road **	12.5
IX	paving-stones	asphalt road	soil sidepath **	concrete sidewalk **	asphalt road	asphalt road **	asphalt road	cobblestone sidewalk stripe	asphalt road	0
X	chipseal	car	truck	car	car	car	car	car	car	0

Table 6. *Cont.*

Row	Prompt	1	2	3	4	5	6	7	8	Hit %
XI	gravel	asphalt road **	asphalt pathway **	asphalt pathway **	sidepath with vegetation	car	asphalt road **	car	wall	0
XII	compacted	truck	car	car	car	car	car	car	car	0
XIII	asphalt	asphalt road	asphalt road	asphalt road	asphalt road	asphalt pathway	asphalt road	asphalt road	asphalt road	100
XIV	concrete plates	truck	nothing	nothing	truck	truck	truck	truck	a stairway	25
XV	ground	asphalt road **	asphalt road **	asphalt road and ground sidewalk	asphalt road and asphalt sidepath	asphalt road **	asphalt road **	asphalt road	asphalt road **	0

**: it was't present on the sample, thus the proper answer would be "none".

We obtained suitable matches only for 'asphalt' and 'grass'. Some categories, such as "sand" and "concrete", were affected by the nature of the test, where random images were selected and often had no correspondence. Although the correct behaviour should be "nothing", this was only seen in the "concrete plates" category. Table 7 presents the "abstract" category results, where a bias towards vehicles was observed. The prompt "anything" yielded more varied answers than "something", while "void" and "nothing" still produced responses, indicating that the language model struggles with abstract concepts.

Table 7. Prompt results for the "abstract" category.

Row	Prompt	1	2	3	4	5	6	7	8	Hit %
I	anything	car	landscape	road	person	road and sky	car panel	vegetation and sky	road and car panel	N/A
II	nothing	asphalt road and noise	asphalt road and noise	car	asphalt road and noise	noisy asphalt road and sky	sky and noise	truck	road, wall, and noise	N/A
III	something	car hood	car panel	traffic signal	car	car	nothing	truck	truck	N/A
IV	anything	road, sidewalk, and vegetation	noisy asphalt road and noisy sky	car	tree	landscape and noisy road	nothing	car	building	N/A
V	void	asphalt road	car	car	nothing	car	car	car	car	N/A

Once again, the bias towards vehicles can be observed. Somehow, "anything" gives more varied answers than "something". "Void" and "nothing" still produce matches, so the language model is not handling this abstract concept correctly. Figure 4 summarises the initial experiment's results, showing the distribution of detected classes with a hierarchical categorization approach similar to that outlined in Figure 3. The most frequently identified classes are "road" and "vehicle", with "sidepath" as the third most common, followed by categories, like "vegetation" and "amalgamation". While "sidepath" appears prominently, there is a possible bias toward road- and vehicle-related features in the dataset, which may reflect the characteristics of the urban environment sampled.

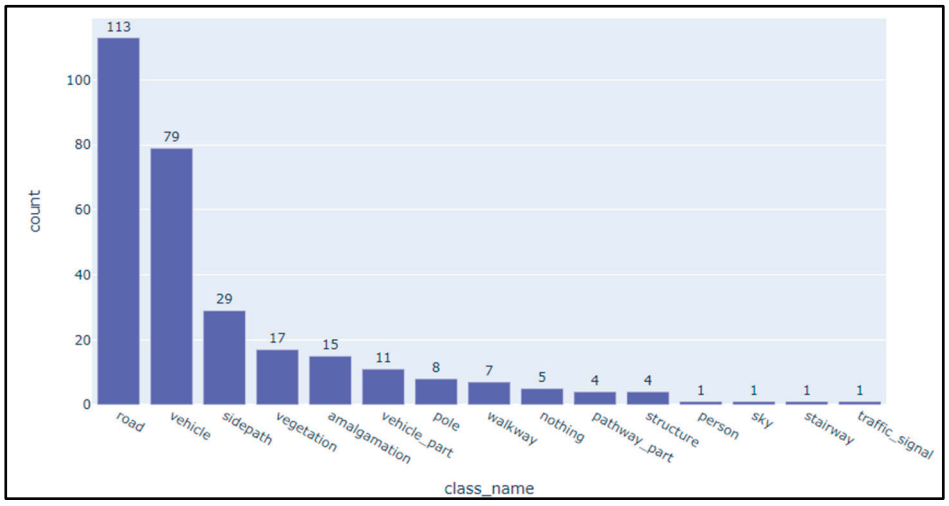


Figure 4. Free prompting and random imagery evaluation summarized results.

3.2. Evaluation of Open-Vocabulary Algorithms for Standardized Pavement Classification

We mainly relied on confusion matrices for the second experiment to summarize the algorithms' performances. In Figure 5, we computed those confusion matrices for all the presented versions.

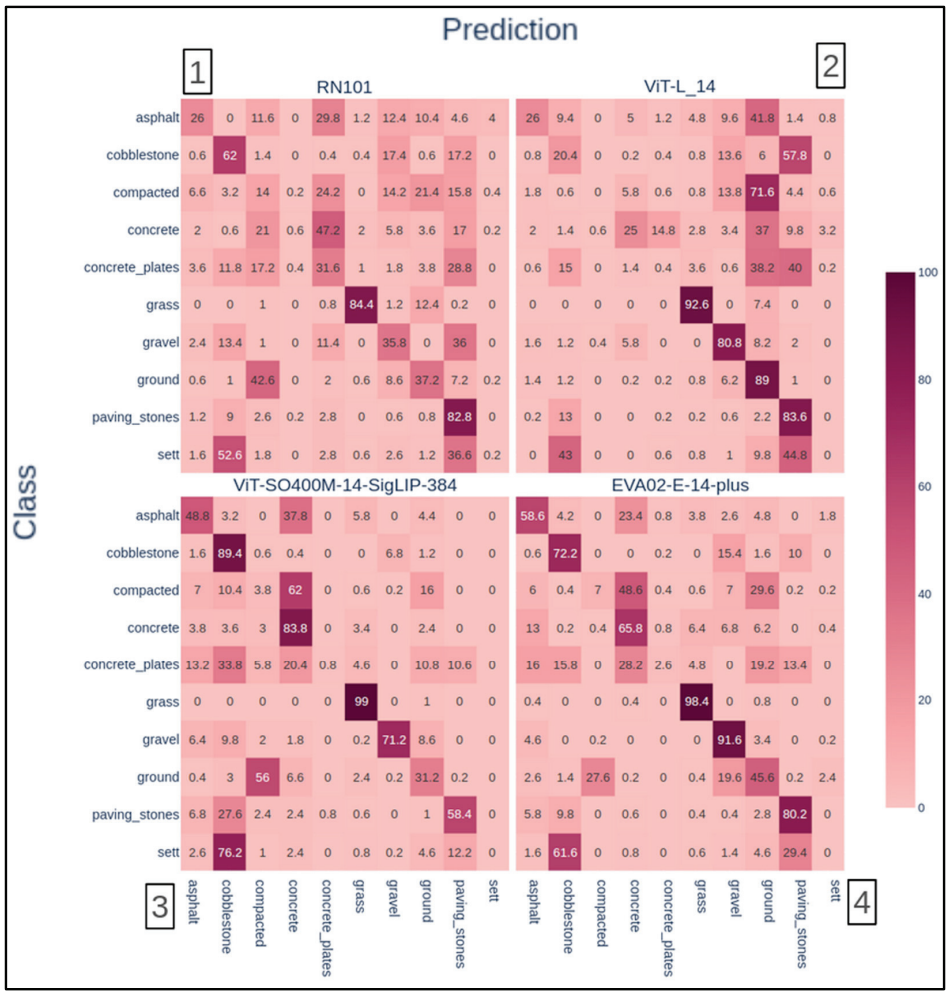


Figure 5. *Cont.*

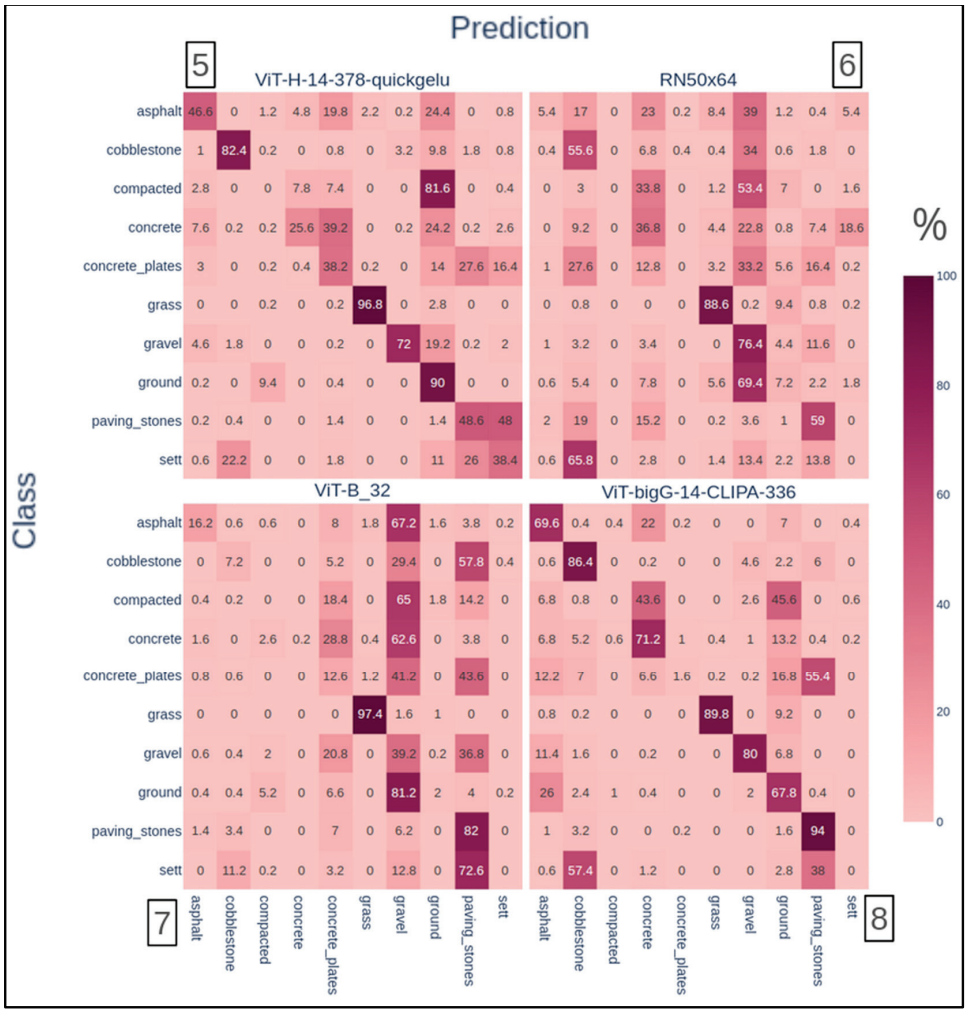


Figure 5. The resulting confusion matrices for the variants of the algorithm.

All true positives are located on the main diagonal in the confusion matrices, with the rows summing to 100%. The columns reveal biases towards certain classes, such as the "Sett" category, which had notably low prediction rates (e.g., only 38% true positives in model 5 and 0% in model 3). Extreme cases were observed in model 7, where the "gravel" class summed to 406% and the "ground" class summed to 318%, indicating significant misclassification issues. It is worth recalling that different datasets were used in each model's training. Therefore, both extremes can be due to imbalanced datasets. In this context, despite their differences, all models are variations of an algorithm that claims to be widely capable of generalizing correctly both in the vision and language domains. It is not only far from happening but also occurs in very different ways throughout all the models.

3.3. Model Aggregation and Fine-Tuning

The results of the aggregation strategy, shown in Figure 6, were obtained by summing the class probabilities from all models to create a composite arbitrary score for each class. The class with the highest score was then selected as the final prediction. This score is used only as an internal ranking mechanism for the purpose of classification and does not represent probabilistic values, as the summed values may exceed one. This test was conducted using all models and the three best overall performers.



Figure 6. Confusion matrices for the aggregation strategy.

While a general improvement was observed, some classes still showed overlap: for instance, "sett" overlapped with "cobblestone" and "paving stones", "concrete plates" overlapped with "paving stones" and "ground", and "compacted" overlapped with "concrete" and "ground". Some classes are still completely hindered, mainly "sett", "concrete plates", and "compacted".

In the fine-tuning experiments (Figure 7), we adjusted various optimizer hyperparameters, as detailed in Table 8, to refine the model performance. The fine-tuning and model selection were carried out empirically, reflecting the complexity of AI models and the absence of a universal closed-form solution [55]. As stated in the methodology, we selected the model ViT-B/32 (number 7), as it obtained the worst overall performance, as shown in Figure 8.

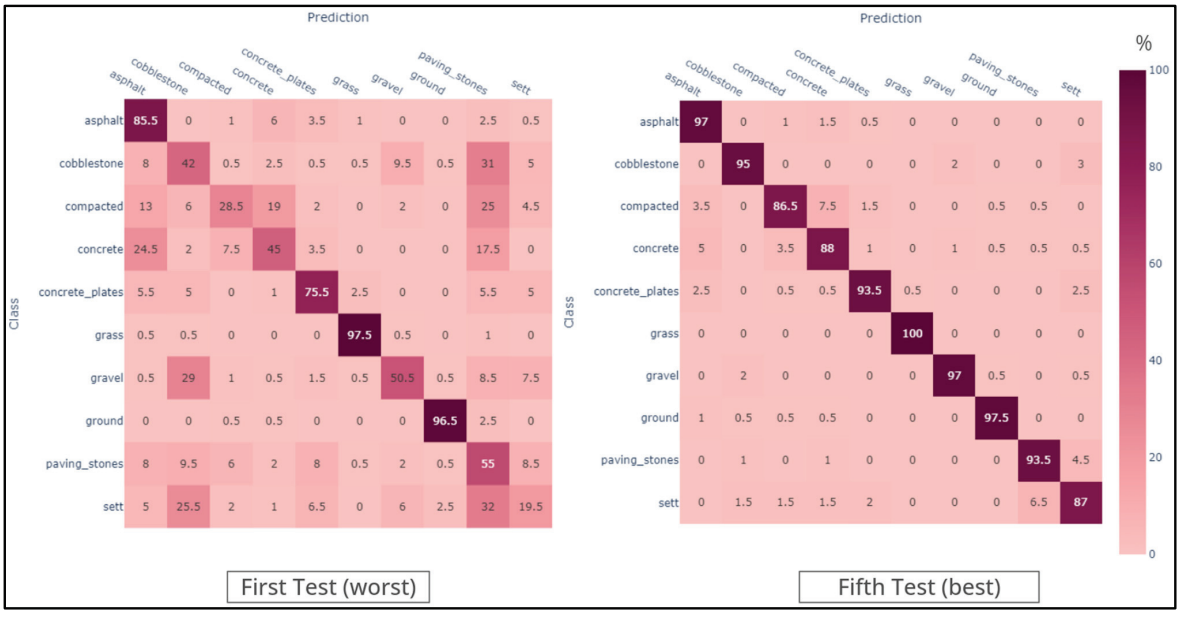


Figure 7. Confusion matrices for the fine-tuning strategy.

Table 8. Hyperparameter variation in the fine-tuning results.

Hyperparameter	Original	Best Result
Batch size	256	320
Learning rate	5.00×10^{-5}	5.00×10^{-6}
Weight decay	0.2	0.5
AMSGrad method	Deactivated	Activated
Betas	(0.9,0.98)	(0.9,0.98)
Epsilon	1.00×10^{-6}	1.00×10^{-6}
Training epochs	100	100

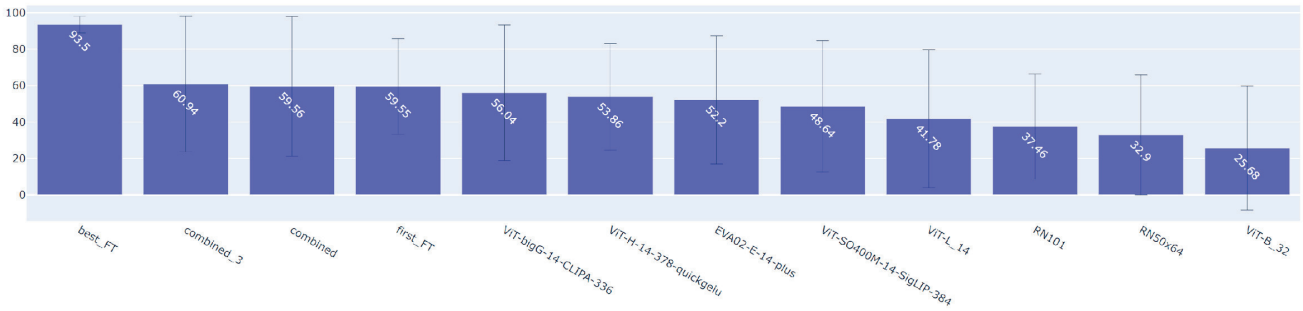


Figure 8. Average TPR of the different variants of the algorithm.

Figure 7 shows that the improvement was quite prominent after reducing the learning rate by a whole order of magnitude, which allowed for better avoidance of local minima, as foretold in the literature [56]. The use of the AMS grad [57] variant also resulted in adaptive capabilities, enabling the variation in the learning rate when mathematically viable. It performs slightly better than the state-of-the-art performance, which, as of April 2024, is an average TPR (true positive rate) of 92.4% on ImageNet [58], according to [59]. It is worth recalling that the best improvement comparison of the final result is with image 5, confusion matrix number 7, which was the starting spot. To compare all the tested scenarios directly, we created the chart in Figure 8, which contains the average TPR and the standard deviation for each tested variation.

The chart in Figure 8 was created to directly compare all tested scenarios, containing the average TPR and the standard deviation for each tested variation. Enforcing a specific meaning for a word with diverse meanings, such as "compacted", played a central role in achieving these groundbreaking results, which are also very consistent among classes. Additionally, defining "sett" as quarried and roughly parallelepipedal bricks was essential, as image searches for "cobblestone" often yielded results that matched this description, reflecting an algorithmic bias corroborated by Figures 5 and 6. These results mostly corroborate the expected that was established on the ontology, with correlations generally appearing on categories with 0 degrees of semantic separation, such as "sett" and "paving stones". One important exception was between the "compacted" and "concrete" categories, in which the proposed ontology fails, and this can be explained by the smoothness frequently present on the bitmaps belonging to these two classes.

In summary, the first experiment's "Free Prompting and Random Imagery Evaluation" demonstrated that while the desired results were generally achieved, the algorithm struggled with hallucinations when targeted features were absent. This result was particularly evident in the "sidepaths" and "surface" categories, where a bias towards detecting roads was observed. The second experiment, "Evaluation of Open-Vocabulary Algorithms for Standardized Pavement Classification", highlighted the strengths and weaknesses of different CLIP algorithm variants. The confusion matrices revealed biases towards certain classes and the challenges in accurately classifying less common pavement types. Both the aggregation and fine-tuning strategies led to improvements, though some classes remained difficult to distinguish. These findings underscore the importance of diverse training datasets and the potential for further enhancements in open-vocabulary models.

4. Conclusions

This study makes key contributions to geospatial analysis and urban infrastructure management. Firstly, we created the Deep Pavements dataset, a robust and expandable dataset specifically designed for training models in standardized (OSM-compatible) pavement material classification. This open dataset is a valuable resource for future research, enabling more comprehensive training and benchmarking of algorithms in this specialized area.

Secondly, we developed a fine-tuned, lightweight CLIP-based model optimized for pavement type detection. This model demonstrates how adapting large, state-of-the-art language models to specific, real-world applications can improve accuracy and efficiency. It also highlights the potential of using LLMs for tasks that require nuanced understanding and classification, offering a more intuitive approach to complex geospatial data challenges.

While this study provides significant insights, there are limitations that highlight opportunities for future research. The model's performance could benefit from incorporating preprocessing techniques to handle low-quality or blurry images, which are common in real-world data. Automating such preprocessing steps would make the pipeline more robust and better suited to large-scale applications. Additionally, our study focused solely on terrestrial imagery. Future research could explore the integration of satellite or sensor data to improve model accuracy and add complementary perspectives, particularly for broader urban infrastructure analysis. Further investigation into optimal train–test split ratios could also provide additional performance insights.

Our findings emphasize the importance of diversifying training datasets to improve the performance of open-vocabulary models in specialized tasks, like pavement type detection. This result aligns with broader challenges in AI, where the reliance on widely used datasets may lead to skewed or limited results in less common applications. Addressing these biases by creating more diverse and representative datasets can significantly enhance model performance and reliability.

Future research should focus on enhancing the CLIP-based model by incorporating different or additional approaches, such as data augmentation and fine-tuning with heavier versions, to compare their performance on generic tasks. Exploring robust solutions that combine specialized CLIP models with constrained vocabularies to verify hallucinations independently could further enhance the reliability of these models. Achieving a truly generalist open-vocabulary algorithm for specialized tasks, like pavement type detection, is an ambitious goal. Still, it is essential for advancing the capabilities of AI in diverse and practical applications.

In conclusion, this study contributes to developing specialized AI models for urban planning and geospatial analysis and provides valuable insights into the broader implications of dataset biases and model generalization. These contributions lay the groundwork for future advancements in creating more flexible, accurate, and representative AI models for real-world applications.

Supplementary Materials: The deep pavements dataset is widely available at https://github.com/kauevestena/deep_pavements_dataset [45], accessed on 1 May 2024; the fine-tuned CLIP-based model is widely available at https://huggingface.co/kauevestena/clip-vit-base-patch32-finetuned-surface-materials [54], accessed on 1 May 2024.

Author Contributions: Conceptualization, Kauê de Moraes Vestena, Silvana Phillipi Camboim, Maria Antonia Brovelli, and Daniel Rodrigues dos Santos; Data curation, Kauê de Moraes Vestena; Formal analysis, Kauê de Moraes Vestena; Funding acquisition, Silvana Phillipi Camboim, and Maria Antonia Brovelli; Investigation, Silvana Phillipi Camboim; Methodology, Kauê de Moraes Vestena, Silvana Phillipi Camboim, Maria Antonia Brovelli, and Daniel Rodrigues dos Santos; Project administration, Silvana Phillipi Camboim; Resources, Silvana Phillipi Camboim, and Maria Antonia Brovelli; Software, Kauê de Moraes Vestena; Supervision, Silvana Phillipi Camboim, Maria Antonia Brovelli, and Daniel Rodrigues dos Santos; Validation, Kauê de Moraes Vestena; Visualization, Kauê de Moraes Vestena; Writing—original draft, Kauê de Moraes Vestena; Writing—review and editing,

Silvana Phillipi Camboim, Maria Antonia Brovelli, and Daniel Rodrigues dos Santos. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financed in part by the Coordination for the Improvement of Higher Education Personnel—Brazil (CAPES)—Finance Code 001.

Data Availability Statement: The Deep Pavements dataset is available under an MIT license: https://github.com/kauevestena/deep_pavements_dataset/blob/main/LICENSE, accessed on 1 May 2024. The fine-tuned CLIP-based model is subject to original OPENAI's licensing terms, with an MIT-based license: https://github.com/openai/CLIP/blob/main/LICENSE, accessed on 1 May 2024. Data are contained within the Supplementary Materials.

Acknowledgments: We acknowledge the Politecnico di Milano's GeoLAB, where most of the present research was physically developed.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hamim, O.F.; Kancharla, S.R.; Ukkusuri, S. Mapping Sidewalks on a Neighborhood Scale from Street View Images. *Environ. Plan. B Urban Anal. City Sci.* **2023**, *51*, 823–838. [CrossRef]
2. Serna, A.; Marcotegui, B. Urban Accessibility Diagnosis from Mobile Laser Scanning Data. *ISPRS J. Photogramm. Remote Sens.* **2013**, *84*, 23–32. [CrossRef]
3. de Moraes Vestena, K.; Camboim, S.P.; dos Santos, D.R. OSM Sidewalkreator: A QGIS Plugin for an Automated Drawing of Sidewalk Networks for OpenStreetMap. *Eur. J. Geogr.* **2023**, *14*, 66–84. [CrossRef]
4. Wood, J. Sidewalk City: Remapping Public Spaces in Ho Chi Minh City. *Geogr. Rev.* **2016**, *108*, 486–488. [CrossRef]
5. Zhou, Z.; Lin, Y.; Li, Y. Large Language Model Empowered Participatory Urban Planning. *arXiv* **2024**, arXiv:2402.17161.
6. Nadkarni, P.M.; Ohno-Machado, L.; Chapman, W.W. Natural Language Processing: An Introduc-Tion. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 544–551. [CrossRef]
7. Wang, X.; Ji, L.; Yan, K.; Sun, Y.; Song, R. Expanding the Horizons: Exploring Further Steps in Open-Vocabulary Segmentation. In *Pattern Recognition and Computer Vision*; Liu, Q., Wang, H., Ma, Z., Zheng, W., Zha, H., Chen, X., Wang, L., Ji, R., Eds.; Springer Nature: Singapore, 2024; pp. 407–419.
8. Eichstaedt, J.C.; Kern, M.L.; Yaden, D.B.; Schwartz, H.A.; Giorgi, S.; Park, G.; Hagan, C.A.; Tobolsky, V.A.; Smith, L.K.; Buffone, A.; et al. Closed- and Open-Vocabulary Approaches to Text Analysis: A Review, Quantitative Comparison, and Recommendations. *Psychol. Methods* **2021**, *26*, 398–427. [CrossRef]
9. Zhu, C.; Chen, L. A survey on open-vocabulary detection and segmentation: Past, present, and future. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 8954–8975. [CrossRef]
10. Zareian, A.; Dela Rosa, K.; Hu, D.H.; Chang, S. Open-Vocabulary Object Detection Using Captions. *arXiv* **2020**, arXiv:2011.10678. Available online: https://arxiv.org/abs/2011.10678 (accessed on 1 May 2024).
11. Yang, G.; Ye, Z.; Zhang, R.; Huang, K. A Comprehensive Survey of Zero-Shot Image Classification: Methods, Implementation, and Fair Evaluation. *Appl. Comput. Intell.* **2022**, *2*, 1–31. [CrossRef]
12. Lampert, C.H.; Nickisch, H.; Harmeling, S. Attribute-Based Classification for Zero-Shot Visual Object Cat-Egorization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 453–465. [CrossRef] [PubMed]
13. Rohrbach, M.; Stark, M.; Schiele, B. Evaluating Knowledge Transfer and Zero-Shot Learning in a Large-Scale Setting. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; IEEE: New York, NY, USA, 2011; pp. 1641–1648.
14. Zhang, D.; Han, J.; Cheng, G.; Yang, M.-H. Weakly Supervised Object Localisation and Detection: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5866–5885. [CrossRef]
15. Vo, H.V.; Siméoni, O.; Gidaris, S.; Bursuc, A.; Pérez, P.; Ponce, J. Active Learning Strategies for Weakly-Supervised Object Detection. In *European Conference on Computer Vision*; Springer Nature: Cham, Switzerland, 2022.
16. Blasiis, M.D.; Benedetto, A.; Fiani, M. Mobile Laser Scanning Data for the Evaluation of Pavement Surface Distress. *Remote Sens.* **2020**, *12*, 942. [CrossRef]
17. Praticò, F.; Vaiana, R. A Study on the Relationship between Mean Texture Depth and Mean Profile Depth of Asphalt Pavements. *Constr. Build. Mater.* **2015**, *101*, 72–79. [CrossRef]
18. Fidalgo, C.D.; Santos, I.M.; Nogueira, C.d.A.; Portugal, M.C.S.; Martins, L.M.T. Urban Sidewalks, Dysfunction and Chaos on the Projected Floor. The Search for Accessible Pavements and Sustainable Mobility. In Proceedings of the 7th International Congress on Scientific Knowledge, Virtual, 20–24 September 2021.
19. Vaitkus, A.; Andriejauskas, T.; Šernas, O.; Čygas, D.; Laurinavičius, A. Definition of concrete and composite precast concrete pavements texture. *Transport* **2019**, *34*, 404–414. [CrossRef]
20. Zeng, Z.; Boehm, J. Exploration of an Open Vocabulary Model on Semantic Segmentation for Street Scene Imagery. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 153. [CrossRef]

21. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
22. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision Meets Robotics: The KITTI Dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
23. Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; Lopez, A.M. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3234–3243.
24. Yu, H.; Yang, Z.; Tan, L.; Wang, Y.; Sun, W.; Sun, M.; Tang, Y. Methods and Datasets on Semantic Seg-Mentation: A Review. *Neurocomputing* **2018**, *304*, 82–103. [CrossRef]
25. Hao, S.; Zhou, Y.; Guo, Y. A brief survey on semantic segmentation with deep learning. *Neurocomputing* **2020**, *406*, 302–321. [CrossRef]
26. Mo, Y.; Wu, Y.; Yang, X.; Liu, F.; Liao, Y. Review the State-of-the-Art Technologies of Semantic Segmenta-Tion Based on Deep Learning. *Neurocomputing* **2022**, *493*, 626–646. [CrossRef]
27. Zou, J.; Guo, W.; Wang, F. A Study on Pavement Classification and Recognition Based on VGGNet-16 Transfer Learning. *Electronics* **2023**, *12*, 3370. [CrossRef]
28. Zhang, C.; Nateghinia, E.; Miranda-Moreno, L.F.; Sun, L. Pavement Distress Detection Using Convolu-Tional Neural Network (CNN): A Case Study in Montreal, Canada. *Int. J. Transp. Sci. Technol.* **2022**, *11*, 298–309. [CrossRef]
29. Riid, A.; Lõuk, R.; Pihlak, R.; Tepljakov, A.; Vassiljeva, K. Pavement Distress Detection with Deep Learning Using the Orthoframes Acquired by a Mobile Mapping System. *Appl. Sci.* **2019**, *9*, 4829. [CrossRef]
30. Mesquita, R.; Ren, T.I.; Mello, C.; Silva, M. Street Pavement Classification Based on Navigation through Street View Imagery. *Ai Soc.* **2022**, *39*, 1009–1025. [CrossRef]
31. Hosseini, M.; Miranda, F.; Lin, J.; Silva, C.T. CitySurfaces: City-scale semantic segmentation of sidewalk materials. *Sustain. Cities Soc.* **2022**, *79*, 103630. [CrossRef]
32. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models from Natural Language Supervision. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021.
33. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Li, C.; Yang, J.; Su, H.; Zhu, J.; et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv* **2023**, arXiv:2303.05499.
34. Grinberger, A.Y.; Minghini, M.; Juhász, L.; Yeboah, G.; Mooney, P. OSM Science—The Academic Study of the OpenStreetMap Project, Data, Contributors, Community, and Applications. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 230. [CrossRef]
35. Zeng, Y.; Huang, Y.; Zhang, J.; Jie, Z.; Chai, Z.; Wang, L. Investigating Compositional Challenges in Vision-Language Models for Visual Grounding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle WA, USA, 17–21 June 2024; pp. 14141–14151.
36. Rajabi, N.; Kosecka, J. Q-GroundCAM: Quantifying Grounding in Vision Language Models via GradCAM. *arXiv* **2024**, arXiv:2404.19128.
37. Wang, S.; Kim, D.; Taalimi, A.; Sun, C.; Kuo, W. Learning Visual Grounding from Generative Vision and Language Model. *arXiv* **2024**, arXiv:2407.14563.
38. Quarteroni, S.; Dinarelli, M.; Riccardi, G. Ontology-Based Grounding of Spoken Language Understanding. In Proceedings of the 2009 IEEE Workshop on Automatic Speech Recognition & Understanding, Moreno, Italy, 13–17 December 2009; IEEE: New York, NY, USA, 2009; pp. 438–443.
39. Baldazzi, T.; Bellomarini, L.; Ceri, S.; Colombo, A.; Gentili, A.; Sallinger, E. Fine-Tuning Large Enterprise Language Models via Ontological Reasoning. In *International Joint Conference on Rules and Reasoning*; Springer Nature: Cham, Switzerland, 2023.
40. Jullien, M.; Valentino, M.; Freitas, A. Do Transformers Encode a Foundational Ontology? Probing Abstract Classes in Natural Language. *arXiv* **2022**, arXiv:2201.10262.
41. Larionov, D.; Shelmanov, A.; Chistova, E.; Smirnov, I. Semantic Role Labeling with Pretrained Language Models for Known and Unknown Predicates. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), Varna, Bulgaria, 2–4 September 2019; Incoma Ltd.: Shoumen, Bulgaria, 2019; pp. 619–628.
42. Smith, M.K.; Welty, C.; McGuinness, D.L. *OWL Web Ontology Language Guide*; W3C: Cambridge, MA, USA, 2004.
43. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023.
44. Meta Platforms, Inc. Mapillary. 2024. Available online: https://www.mapillary.com/ (accessed on 20 November 2024).
45. Vestena, K. GitHub—Kauevestena/deep_pavements_dataset. GitHub. Available online: https://github.com/kauevestena/deep_pavements_dataset (accessed on 20 November 2024).
46. Fan, Q.; Tao, X.; Ke, L.; Ye, M.; Zhang, Y.; Wan, P.; Wang, Z.; Tai, Y.-W.; Tang, C.-K. Stable Segment Anything Model. *arXiv* **2023**, arXiv:2311.15776.
47. Hetang, C.; Xue, H.; Le, C.; Yue, T.; Wang, W.; He, Y. Segment Anything Model for Road Network Graph Extraction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle WA, USA, 17–21 June 2024.

48. Son, J.; Jung, H. Teacher–Student Model Using Grounding DINO and You Only Look Once for Multi-Sensor-Based Object Detection. *Appl. Sci.* **2024**, *14*, 2232. [CrossRef]
49. Dong, X.; Bao, J.; Zhang, T.; Chen, D.; Gu, S.; Zhang, W.; Yuan, L.; Chen, D.; Wen, F.; Yu, N. CLIP Itself Is a Strong Fine-Tuner: Achieving 85.7% and 88.0% Top-1 Accuracy with ViT-B and ViT-L on ImageNet. *arXiv* **2022**, arXiv:2212.06138.
50. Nguyen, T.; Ilharco, G.; Wortsman, M.; Oh, S.; Schmidt, L. Quality Not Quantity: On the Interaction between Dataset Design and Robustness of Clip. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 21455–21469.
51. Fang, A.; Ilharco, G.; Wortsman, M.; Wan, Y.; Shankar, V.; Dave, A.; Schmidt, L. Data Determines Distributional Robustness in Contrastive Language Image Pre-Training (Clip). In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 6216–6234.
52. Tu, W.; Deng, W.; Gedeon, T. A Closer Look at the Robustness of Contrastive Language-Image Pre-Training (Clip). *Adv. Neural Inf. Process. Syst.* **2024**. [CrossRef]
53. Mumuni, F.; Mumuni, A. Segment Anything Model for Automated Image Data Annotation: Empirical Studies Using Text Prompts from Grounding DINO. *arXiv* **2024**, arXiv:2406.19057.
54. Kaue-Vestena/Clip-Vit-Base-Patch32-Finetuned-Surface-Materials. Hugging Face. 2024. Available online: https://huggingface.co/kauevestena/clip-vit-base-patch32-finetuned-surface-materials (accessed on 20 November 2024).
55. Eimer, T.; Lindauer, M.; Raileanu, R. Hyperparameters in Reinforcement Learning and How To Tune Them. In Proceedings of the 40th International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023.
56. Tong, Q.; Liang, G.; Bi, J. Calibrating the Adaptive Learning Rate to Improve Convergence of ADAM. *Neurocomputing* **2022**, *481*, 333–356. [CrossRef]
57. Reddi, S.J.; Kale, S.; Kumar, S. On the Convergence of Adam and Beyond. *arXiv* **2019**, arXiv:1904.09237.
58. Yu, J.; Wang, Z.; Vasudevan, V.; Yeung, L.; Seyedhosseini, M.; Wu, Y. CoCa: Contrastive Cap-Tioners Are Image-Text Foundation Models. *arXiv* **2022**, arXiv:2205.01917.
59. Code, P.W. Papers with Code—ImageNet Benchmark (Image Classification). *GitHub*. 2024. Available online: https://paperswithcode.com/sota/image-classification-on-imagenet?metric=GFLOPs (accessed on 20 November 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

MDPI Books Editorial Office
E-mail: books@mdpi.com
www.mdpi.com/books



Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Topic Editors. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editors and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.





Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-9184-9