Published in Journals: Applied Sciences, Electronics, Remote Sensing and AI

Topic Reprint

Computational Intelligence in Remote Sensing

Volume I

Edited by
Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

mdpi.com/topics

MDPI

Computational Intelligence in Remote Sensing—Volume I

Computational Intelligence in Remote Sensing—Volume I

Editors

Yue Wu
Kai Qin
Maoguo Gong
Qiguang Miao

MDPI

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

Editors
Yue Wu
Xidian University
Xi'an
China

Kai Qin
Swinburne University of Technology
Hawthorn, VIC
Australia

Maoguo Gong
Xidian University
Xi'an
China

Qiguang Miao
Xidian University
Xi'an
China

Editorial Office
MDPI
St. Alban-Anlage 66
4052 Basel, Switzerland

This is a reprint of articles from the Topic published online in the open access journals *Applied Sciences* (ISSN 2076-3417), *Electronics* (ISSN 2079-9292), and *Remote Sensing* (ISSN 2072-4292) (available at: https://www.mdpi.com/topics/Remote_Sensing).

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.

Volume I
ISBN 978-3-7258-0411-5 (Hbk)
ISBN 978-3-7258-0412-2 (PDF)
doi.org/10.3390/books978-3-7258-0412-2

Set
ISBN 978-3-7258-0377-4 (Hbk)
ISBN 978-3-7258-0378-1 (PDF)

Contents

remote sensing

MDPI

Editorial

Computational Intelligence in Remote Sensing

Yue Wu [1,*], Maoguo Gong [2], Qiguang Miao [1] and Kai Qin [3]

[1] Department of Computer Science and Technology, Xidian University, Xi'an 710071, China; qgmiao@xidian.edu.cn
[2] Key Laboratory of Intelligent Perception and Image Understanding, Xidian University, Xi'an 710071, China; gong@ieee.org
[3] Department of Computer Science and Software Engineering, Swinburne University of Technology, Victoria 3122, Australia; kqin@swin.edu.au
* Correspondence: ywu@xidian.edu.cn

1. Introduction

With the development of Earth observation techniques, vast amounts of remote sensing data with a high spectral–spatial–temporal resolution are captured all the time, and remote sensing data processing and analysis have been successfully used in numerous fields, including geography, environmental monitoring, land survey, disaster management, mineral exploration and more. For the processing, analysis and application of remote sensing data, there are many challenges, such as the vast amount of data, complex data structures, small labeled samples and nonconvex optimization. In recent years, the convergence of computational intelligence (CI) and remote sensing has ushered in a new era of possibilities for understanding and harnessing the wealth of information that Earth observation satellites provide. Computational intelligence methods, such as deep neural networks, evolutionary optimization and swarm intelligence, have demonstrated remarkable capabilities in unveiling intricate patterns within satellite images, time series data and multispectral/hyperspectral information. In the future, CI will produce effective solutions to the challenges in remote sensing.

2. Recent Research and Progress

This Topic series aims to highlight the latest research and advances in the application of computational intelligence in the field of remote sensing. In total, this Topic series contains 12 papers written by research experts on topics of interest. Based on the synthesis of these latest achievements, they can be categorized into four sections: computational intelligence methods in hyperspectral remote sensing images; object detection techniques in remote sensing images; deep learning approaches in remote sensing image classification and intelligent optimization and control in satellite image applications.

2.1. Computational Intelligence Methods in Hyperspectral Remote Sensing Images

This section consists of three papers. The first paper is written by A.C.P. Silva, K.T.Z. Coimbra, L.W.R. Filho, G. Pessin and R.E. Correa-Pabón. They mainly explore the possibility of applying machine learning models to monitor the quality of iron ore [1]. The second paper, written by W. Shuai, F. Jiang, H. Zheng and J. Li, mainly proposes a new method with high processing efficiency for change detection in remote sensing images, called MSGATN [2]. The last work studies SAR image segmentation based on fuzzy c-means and is by J. Zhu, F. Wang and H. You. Experiments show that the framework can achieve more than 97% segmentation accuracy [3].

2.2. Object Detection Techniques in Remote Sensing Images

The following three papers mainly utilize deep learning techniques to solve practical problems in the field of remote sensing image object detection. The first paper,

Citation: Wu, Y.; Gong, M.; Miao, Q.; Qin, K. Computational Intelligence in Remote Sensing. *Remote Sens.* **2023**, *15*, 5325. https://doi.org/10.3390/rs15225325

Received: 23 October 2023
Accepted: 24 October 2023
Published: 12 November 2023

by R. Chen and S. Liu et al., proposes an effective infrared object detection method based on source model guidance [4]. They show two explicit examples based on CenterNet and YOLOv3, respectively, and experimentally demonstrate that the method can achieve powerful performance with limited samples. The second paper, by L. Yu and X. Zhou et al., proposes a method for boundary-aware salient object detection in optical remote sensing images [5]. The method uses a graph convolutional network-based feature extraction module and a boundary-aware attention-based module to improve the accuracy and robustness of boundary-aware salient object detection. The third paper, by F. Zhou and H. Deng et al., studies deep learning-based aircraft detection [6]. The paper proposes an enhanced YOLOv5 model in which a ConvNext-based feature extraction module and a Transformer-based feature fusion module are used to improve the detection performance.

2.3. Deep Learning Approaches in Remote Sensing Image Classification

This section includes three papers. The first paper is authored by H. Toriya and A. Dewan et al., who primarily explore the key point matching problem in image features. They propose using a deep neural network (DNN) to construct an image translator and introduce a new edge enhancement filter methodology within the conditional generative adversarial network (cGAN) structure to tackle this issue [7]. The second paper, written by Z. Wei and Z. Zhang, describes a network built on multi-level strip pooling and a feature enhancement module (MSPFE-Net). Here, deep learning is effectively applied to address the challenge of road extraction [8]. In the third paper, L. Zeng and Y. Huo et al. develop the high-quality seed instance mining (HSIM) module, alongside the dynamic pseudo-instance label assignment (DPILA), to address the issue of weakly supervised detection in remote sensing images [9].

2.4. Intelligent Optimization and Control in Satellite Image Applications

This section includes three state-of-the-art papers for reference focusing on different research directions in satellite images. The first paper is authored by T. Zheng, Y. Dai, C. Xue and L. Zhou. They propose a method for solving near-lossless hyperspectral data compression using recursive least squares. They use the linear combination of previous pixels to predict the target pixel values while using a recursive least squares filter to iteratively update the weight matrix for prediction, which effectively removes spatial and spectral redundancy information [10]. The second paper is written by N. Andrijević, V. Urošević, B. Arsić, D. Herceg and B. Savić. This paper designs a time prediction model for bee influx and outflow in a bee colony ecosystem with a large number of sensors by simulating the correlation between the environment and bee colony activity to simulate the bee colony ecosystem [11]. L. Li, D. Yin, Q. Li, Q. Zhang and Z. Mao propose a verification method for ultraviolet imagers using the seeker optimization algorithm. This method can effectively use ultraviolet imagers to conduct authenticity check studies on ocean surface radiation data [12].

3. Discussion

The papers provide an exchange platform for researchers in the field of remote sensing images, covering topics such as hyperspectral remote sensing image processing, remote sensing image classification, segmentation, object detection and intelligent optimization and control in satellite image applications. These themes represent a series of key issues in the field of remote sensing images. The research papers in this journal not only delve into these issues, but also propose new methods and ideas, providing strong support for future research directions.

In this issue of the journal, we have seen a series of important developments in the field of hyperspectral remote sensing image processing. Researchers have utilized the rich information of hyperspectral data to not only improve the performance of segmentation, but also provide new tools for application fields such as resource management and environmental monitoring. In addition, remote sensing image classification, segmentation and

object detection have always been research hotspots. Research in this journal shows that deep learning technology has made significant progress in the application of these tasks.

The papers in this research Topic showcase the innovative and influential contributions of researchers in this field. Researchers have not only delved into various issues, but also proposed many new methods and technologies, demonstrating the potential of computational intelligence in advancing our understanding of remote sensing images and providing strong support for future research directions. In the future, we can look forward to more interdisciplinary cooperation, combining remote sensing image research with application fields such as environmental science, agriculture and urban planning to solve complex real-world problems. We encourage readers to further explore the cutting-edge research and novel applications presented in these papers to provide new impetus for scientific and technological innovation.

Author Contributions: Conceptualization, Y.W. and M.G.; writing—original draft preparation, Y.W. and M.G.; writing—review and editing, Q.M. and K.Q. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: Thanks to all authors, peer reviewers and editorial team members for their valuable contributions. Their dedication and hard work have been instrumental in the outcome of this Topic series. Herewith, congratulations to all the authors for their outstanding achievements on relevant topics.

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

References

1. Silva, A.C.P.; Coimbra, K.T.Z.; Filho, L.W.R.; Pessin, G.; Correa-Pabón, R.E. Monitoring of Iron Ore Quality through Ultra-Spectral Data and Machine Learning Methods. *AI* **2022**, *3*, 554–570. [CrossRef]
2. Shuai, W.; Jiang, F.; Zheng, H.; Li, J. MSGATN: A Superpixel-Based Multi-Scale Siamese Graph Attention Network for Change Detection in Remote Sensing Images. *Appl. Sci.* **2022**, *12*, 5158. [CrossRef]
3. Zhu, J.; Wang, F.; You, H. SAR Image Segmentation by Efficient Fuzzy C-Means Framework with Adaptive Generalized Likelihood Ratio Nonlocal Spatial Information Embedded. *Remote Sens.* **2022**, *14*, 1621. [CrossRef]
4. Chen, R.; Liu, S.; Mu, J.; Miao, Z.; Li, F. Borrow from Source Models: Efficient Infrared Object Detection with Limited Examples. *Appl. Sci.* **2022**, *12*, 1896. [CrossRef]
5. Yu, L.; Zhou, X.; Wang, L.; Zhang, J. Boundary-Aware Salient Object Detection in Optical Remote-Sensing Images. *Electronics* **2022**, *11*, 4200. [CrossRef]
6. Zhou, F.; Deng, H.; Xu, Q.; Lan, X. CNTR-YOLO: Improved YOLOv5 Based on ConvNext and Transformer for Aircraft Detection in Remote Sensing Images. *Electronics* **2023**, *12*, 2671. [CrossRef]
7. Toriya, H.; Dewan, A.; Ikeda, H.; Owada, N.; Saadat, M.; Inagaki, F.; Kawamura, Y.; Kitahara, I. Use of a DNN-Based Image Translator with Edge Enhancement Technique to Estimate Correspondence between SAR and Optical Images. *Appl. Sci.* **2022**, *12*, 4159. [CrossRef]
8. Wei, Z.; Zhang, Z. Remote Sensing Image Road Extraction Network Based on MSPFE-Net. *Electronics* **2023**, *12*, 1713. [CrossRef]
9. Zeng, L.; Huo, Y.; Qian, X.; Chen, Z. High-Quality Instance Mining and Dynamic Label Assignment for Weakly Supervised Object Detection in Remote Sensing Images. *Electronics* **2023**, *12*, 2758. [CrossRef]
10. Zheng, T.; Dai, Y.; Xue, C.; Zhou, L. Recursive Least Squares for Near-Lossless Hyperspectral Data Compression. *Appl. Sci.* **2022**, *12*, 7172. [CrossRef]
11. Andrijević, N.; Urošević, V.; Arsić, B.; Herceg, D.; Savić, B. IoT Monitoring and Prediction Modeling of Honeybee Activity with Alarm. *Electronics* **2022**, *11*, 783. [CrossRef]
12. Li, L.; Yin, D.; Li, Q.; Zhang, Q.; Mao, Z. An Exploratory Verification Method for Validation of Sea Surface Radiance of HY-1C Satellite UVI Payload Based on SOA Algorithm. *Electronics* **2023**, *12*, 2766. [CrossRef]

remote sensing

MDPI

Technical Note

Specific Windows Search for Multi-Ship and Multi-Scale Wake Detection in SAR Images

Kaiyang Ding †, Junfeng Yang †, Zhao Wang, Kai Ni , Xiaohao Wang and Qian Zhou *

Division of Advanced Manufacturing, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China; dingky19@mails.tsinghua.edu.cn (K.D.); yjf20@mails.tsinghua.edu.cn (J.Y.); z-wang20@mails.tsinghua.edu.cn (Z.W.); ni.kai@sz.tsinghua.edu.cn (K.N.); xhwang@mail.tsinghua.edu.cn (X.W.)
* Correspondence: zhou.qian@sz.tsinghua.edu.cn; Tel.: +86-15-88962-6087
† These authors contributed equally to this paper.

Abstract: Traditional ship identification systems have difficulty in identifying illegal or broken ships, but the wakes generated by ships can be used as a major feature for identification. However, multi-ship and multi-scale wake detection is also a big challenge. This paper combines the geometric and pixel characteristics of ships and their wakes in Synthetic Aperture Radar (SAR) images and proposes a method for multi-ship and multi-scale wake detection. This method first detects the highlight pixel area in the image and then generates specific windows around the centroid, thereby detecting wakes of different sizes in different areas. In addition, all wake components can be located completely based on wake clustering, the statistical features of wake axis pixels can be used to determine the visible length of the wake. Test results on the Gaofen-3 SAR image show the special potential of the method for wake detection.

Keywords: ship wake; wake detection; specific windows; multi-ship; multi-scale; Gaofen-3

Citation: Ding, K.; Yang, J.; Wang, Z.; Ni, K.; Wang, X.; Zhou, Q. Specific Windows Search for Multi-Ship and Multi-Scale Wake Detection in SAR Images. *Remote Sens.* **2022**, *14*, 25. https://doi.org/10.3390/rs14010025

Academic Editors: Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Received: 23 November 2021
Accepted: 19 December 2021
Published: 22 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

With the development of the marine industry, there are many ports, a huge increase of ships, complicated waterways and rapidly changing sea conditions in marine countries in the world, which greatly increases the risk of ship encounters. Moreover, the exploitation of marine resources has also caused problems such as illegal invasion, illegal fishing and illegal smuggling. Therefore, more and more attention has been paid to marine ship monitoring in the whole sea area. SAR has been widely used in ship detection [1,2], oil spill detection [3], change detection [4–7] and other fields [8,9] and plays an important role in ship detection due to its wide observation range, short observation period, strong data timeliness and high spatial resolution [10]. At present, SAR image ship detection includes ship body detection and ship wake detection. Compared with the detection of ship body alone, the detection of wake is more valuable in terms of detectability and researchability. First, the wake lasts for a long time: under certain conditions, the wake on the sea can stretch over tens of kilometers, which is often dozens of times the length of the ship [11,12]. In addition, wakes generated by ships in different motion states have different geometric and pixel features. Detection of wakes can not only locate ship targets indirectly, but also determine their sailing speed, track, ship type and other information according to the geometric features of the wakes [13–15].

In fact, the linear structure is the main feature of each wake component in the SAR image. These linear features have a certain length and width, and there are narrow regions of bright or dark. Therefore, wake detection can proceed from this linear feature and transform the problem into one of line detection. The Radon transform (RT) or Hough transform (HT) has shown excellent performance in this field [16,17]. In addition, the limited regional characteristics of ship wakes can be transformed into a target detection problem; the rapidly developing deep learning method has also been applied to wake

detection. Kang et al. [18] detected ships as well as wakes from SAR images using deep learning based on Convolutional Neural Networks (CNN). The detection rates under adverse weather conditions were 68.4% and 60.0%, respectively. However, deep learning methods usually require a large amount of data support, and there are not enough open wake data sets hindering the rapid development of this method [19,20]. Therefore, most of the current research is based on the traditional wake detection method of RT or HT. Considering that SAR images are seriously disturbed by speckle noise or clutter, a lot of work is focused on image data preprocessing to make the wake characteristics more obvious. Jin et al. [21] proposed a spatial wavelet correlation technique for ship wake detection. After multi-scale edge extraction and spatial correlation, the wake is extracted effectively and the edge of the wake is sharpened significantly. Courmontagne et al. [22] introduced Stochastic Matched Filtering into wake detection, and Arnord-Bos et al. [23] applied it to maximize the signal-to-noise ratio after processing. Biondi [24] considered the polarization information of SAR images and adopted Low-rank Plus Sparse Decomposition followed by RT to perform clutter suppression and extract the interesting wake components. Yang et al. [25] constructed the wake structure dictionary in an analytical way and decomposed the image into structural components including ship wake and sea texture components, which suppressed the marine clutter noise in a disguised way and had a significant effect on ship wake detection in SAR images with complex backgrounds. Additionally, much research is devoted to improving these traditional methods to make them have better applicability and robustness. Copeland et al. [26] proposed line RT: intensity integration is done on short segments instead of on the whole image, which can detect and locate wakes that are obviously smaller than the image dimension. There are also some scholars who use local RT, or a combination of sliding windows for global wake detection [27,28] so as to realize the detection of local short wake. However, this kind of local processing algorithm often consumes a lot of time and computation power, which is not conducive to the real-time detection of wake. Apart from these methods, the circular scanning method [29], the image energy method [30] and the pixel screening method [31,32] also have good performance in the field of wake detection.

Many of the above algorithms are essentially wake extraction under the condition of known wake. However, the actual situation is often that we cannot know how many ships and wakes are contained in the SAR image, and the scale of the wakes cannot be determined in advance. In real SAR ocean background images, a single image may contain many wakes of different sizes and positions, and the wakes usually occupy a small area. Therefore, most of these algorithms cannot effectively deal with the problem of multi-ship and multi-scale wake detection in unfamiliar images. This article from the ship and its wake pixel features and geometric characteristics, puts forward the Specific Window Search method for wake detection, a series of search windows with different sizes and orientations are generated around the highlighted pixel (ship or other man-made objects) field of the image. Each window is scored based on improved RT to screen out the area containing wake components and determine whether the highlighted pixel point in the center is a ship. Then, the turbulence and Kelvin wakes are located by clustering analysis of the candidate locations based on the geometric characteristics of wakes in the region containing wakes. Finally, we extract the pixels on the main axis of the retrieved wake and determine the beginning and end points of the wake based on statistical analysis characteristics and pixel gradient characteristics. As we know, separately detecting ships or wakes can only locate the position or judge the passage of the ship. Only by paired detection of ship and wake can we make better use of their geometric relationship to conduct parameter inversion research [33,34]. This is also the core of the algorithm, which uses highlighted pixels to quickly locate the wake, and the detected wake lines help to determine whether the center is a ship. We have applied the algorithm to SAR images collected by Gaofen-3 [35] and tested multiple images with different backgrounds and styles in the data set to complete wake line localization and length measurement. The main contributions of this paper are as follows:

1. A specific window search method different from pre-selected box generation and sliding window search is proposed for the multi-ship and multi-scale wake detection problem. Search sub-windows are generated based on a limited number of highlighted pixel regions in the image, thus greatly reducing the area to be detected.
2. Combining the geometric features of the ship and the wake, we develop the correlation detection of the ship and the wake, which are detected in pairs rather than separately, and help in the inversion of the ship navigation information.
3. Based on the angle characteristics between wake components, a new clustering method is proposed to locate different wake components (turbulence and Kelvin wake) of the same ship, and measure the shortest visible length of the wake.
4. We create SAR wake data set containing different types of Gaofen-3 and validate our method on these data.

The rest of the paper is organized as follows. Section 2 introduces in detail the specific window search detection algorithm we propose, including the wake location strategy and wake length measurement method. Section 3 presents the Gaofen-3 SAR data and analyzes the results through several groups of experiments and comparison. Section 4 concludes the paper with a summary and puts forward some suggestions for future development.

2. Materials and Methods

In this section, we will show the ship and its wake characteristics in SAR images, detail our specific window search algorithm for wake detection and propose some strategies for getting as many potential locations as possible. Our search algorithm has the following design considerations:

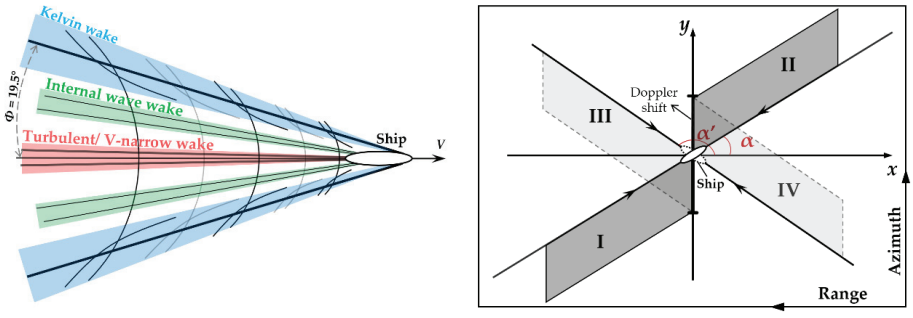
First of all, it is necessary to capture all the ship wakes, which are the identity features generated by the movement of the ship. Accurately identifying all the wake features is the main goal of the algorithm.

Second, the scale of the wakes generated by ships are not the same under different motion states, which are affected by the ship's speed and shape as well as the sea conditions. The acquisition of wakes of all different scales is helpful for the subsequent inversion of ship motion information.

Finally, our algorithm is different from the traditional sliding window search algorithm. The goal of a specific window search is to generate a certain number of windows under a specific target, so as to locate the actual position of the wake. This set size is much smaller than the traditional algorithm, so the efficiency of our algorithm will be greatly improved.

2.1. Specific Windows Search by Highlighted Pixel Points

Wakes generated by ships during navigation can generally be divided into Kelvin wakes, internal wave wakes, turbulent wakes and V-narrow wakes, and they show different geometric characteristics [36], as shown in Figure 1a. Actually, ship wakes in SAR images are mainly turbulent wakes and Kelvin wakes, with Kelvin wakes accounting for about 17% of wakes. Turbulent wakes are the most common wake type and exist in almost all ship wakes [37]. They are characterized by dark or bright narrow lines that stretch for tens of kilometers in length; these ships generally show the highlight point or region.



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

Figure 1. Various wakes of ships and their geometric relations: (**a**) The geometry of the wake pattern produced by ships; (**b**) The relationship between ship direction and wake region: I, II, III and IV represent the wake region caused by the maximum Doppler shift in four directions, α and α′ are the angle of two tracks respectively.

In addition, due to the ship's movement and the SAR system, a Doppler shift effect will appear in the imaging process, resulting in a special geometric relationship between the ship and the wake [38,39], as shown in Figure 1b. In the imaging mode of the ascending left view, the ships travel toward each other in 2 tracks with different angles and 4 different directions. The regions where wakes are generated (shown by turbulent wakes) are divided into four different regions. Taking a ship heading northeast, as an example, the ship target is located in the coordinate center, and its turbulent wake must be generated in area **I**. Therefore, unlike other targets on the sea surface, the pixel features and spatial geometric features of ship targets and wakes are very obvious. It is precisely by combining the pixel and geometric features between the ship and the wake that we use the highlight pixel algorithm to form the basis of our specific window search, so as to avoid irrelevant interference features and improve the specificity of the target search. The core idea of the specific window search is to generate a specific window based on the highlighted pixels for the target search. The steps of specific window generation are as follows:

The first thing we need to do is to detect the location or potential location of the ships. However, in addition to ship targets, artifacts on the sea surface, islands and other speckle noises are also displayed in the highlighted pixel areas. Therefore, we need to do image preprocessing to eliminate the influence of these non-ideal factors. In this way, the real position of the ship can be determined as accurately as possible, and the computational load of the subsequent algorithm can be reduced.

We preset a constant false alarm rate and obtain all potential ship target areas through the pixel filtering algorithm. Then, for the potential target points, the method adopted in this paper is performing morphological processing on the binarization target image to eliminate discrete noise points and enhance the potential ship target area.

After obtaining the position of the potential ship target, according to the geometric relationship between the ship and the wake in the SAR image described above, specific windows are successively generated around the centroid of the highlighted pixel area of the ship target in order to cover different areas where the ship wake exists. Figure 2 shows an example of the specific window, and the corresponding specific window can be generated similarly for multiple wakes.

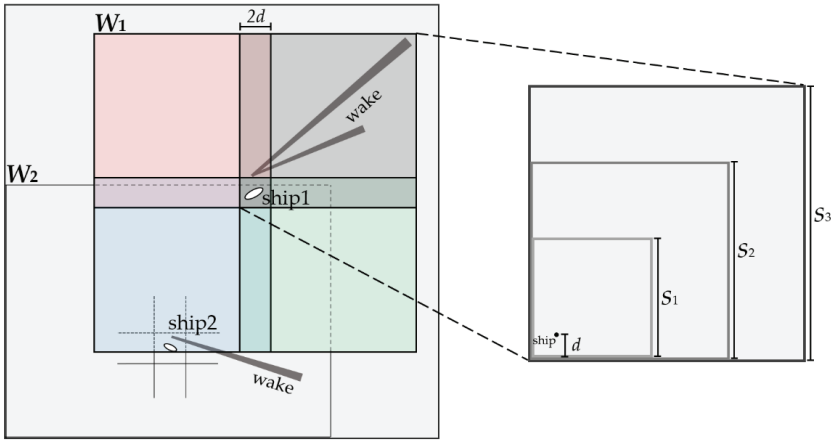


Figure 2. Schematic diagram of Specific Window distribution.

The length of the subwindow is set as a multi-scale window, so that wakes of different lengths can be detected. In addition, to cover the offset d due to the Doppler shift, an overlap width of $2d$ is set between the windows. The offset d can be calculated by [40,41]:

$$d = \frac{v_m R_s \cos\theta_{in}}{V_s} \tag{1}$$

where v_m denotes the maximum velocity of the ship, V_s is velocity of the satellite, R_s is the slant range distance and θ_{in} is the incidence angle. These satellite-related parameters used in the experiment are all from Gaofen-3.

Each detection window is represented by a subwindow in the set $W_i = \{w^i_{j,k}\}$ and the corresponding score $s^i_{j,k}$. The scoring rules will be described in detail below, where i represents the set of windows generated by the i-th pixel $p_i = (x_i, y_i)$, and $(j, k) = (\{S_1, S_2, S_3\}, \{\pm\pi/4, \pm 3\pi/4\})$ determine the scale and orientation of subwindow respectively.

The key problem of the algorithm is to find the location and measure the length of the wake. Therefore, some strategies are designed to make the location and length more accurate.

2.2. Wake Localization Strategy

The size and location of the window covering the object varies in the image. However, within a set of windows in a highlighted pixel area, some windows cover objects more accurately than others. An appropriate window facilitates subsequent standardization of wake features with varying scales. In the wake localization stage, for each subwindow, we evaluated the possibility of a window covering an object by an improved RT, based on its internal pixel integration, while considering its position and size [16]. The localization stage includes first determining the window orientation, that is, finding the window directions that can accurately cover the wake position in the four directions, and then accurately marking the wake position of each scale or each component in the window.

In order to avoid the influence of the highlighted pixel region on the subwindow pixel integration, we mask these regions with the average pixel $\mu(w^i_{j,k})$ of the sea clutter in the window. For the window set W_i, the minimum scale window determines the shortest wake that can be detected, the Local Difference Radon transform in this scale window set w^i_k is used as its score to eliminate the fake ship highlight pixels and determine the window orientation. The probability score s^i_k of each direction window is expressed by:

$$\mathrm{s^i_k} = \max R^{\mathrm{i}}_{\mathrm{k}} \tag{2}$$

where, R_k^i represents the Difference Radon transform. Although the actual ship wake may be bright or dark lines, its average pixel is different from sea clutter; local pixel difference processing can increase the contrast between wake and clutter, and, at the same time, it is convenient to capture the position of the wake at the peak in the Radon domain, which is defined as:

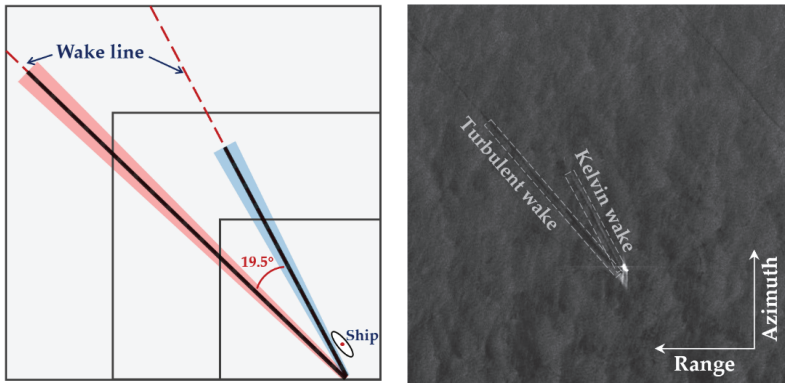
$$R_k^i = \int\int_{w_k^i} \left| f(x,y) - \mu(w_k^i) \right| \delta(\rho - xcos\theta - ysin\theta)\, dxdy \tag{3}$$

The maximum value of s_k^i is taken as the score of the i-th window set. When the root location of the subwindow is the fake highlight pixel centroid without wakes, the score is approximately zero. If the value is much larger than zero, the subwindow is judged to contain wakes. Thus, the subwindow containing the wake can be selected and its orientation can be determined, the corresponding (i, k) is shown in Equation (5).

$$s^i = \max_{k=\pm\pi/4,\pm 3\pi/4} s_k^i \tag{4}$$

$$w_k^i \Leftarrow \left\{ (i,k) \middle| \max(\max R_k^i) \gg 0,\ i = 1,2,\ldots,n,\ k = \pm\pi/4, \pm 3\pi/4 \right\} \tag{5}$$

After completing this step, the wakes in the image are positioned by a series of subwindows of different sizes, as shown in Figure 3a. In addition, due to imaging conditions, different wakes formed by the same ship may show different lengths in the actual SAR image [13,42], as shown in Figure 3b.



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

Figure 3. Fine detection of wakes from the same ship: (**a**) Wakes of different scales and their detection subwindows; (**b**) turbulence wakes and Kelvin wakes of different lengths in SAR images.

In order to accurately locate wakes at various scales and ensure that different wake components generated by the same ship can be detected, we modified Equation (3) by adding scale factors to standardize it; S_j represents window scale.

$$\overline{R}_j^i(\rho,\theta) = R_j^i(\rho,\theta)/Sj \tag{6}$$

We then need to set the appropriate threshold T_w to mark all the qualified position information (ρ,θ); T_w is usually set to $R_m/\sqrt{2}$, which is given by the length relation between the Kelvin wake and the turbulent wake. R_m is the maximum value in the Radon domain.

$$(\rho,\theta) = \arg \overline{R}_j^i(\rho,\theta)) > Tw \tag{7}$$

Since the number of wakes in the subwindow is unknown and the wake component is not a line but a channel of bright/dark pixels, the position set in Equation (7) is some cluster of points, $\{(\rho_1,\theta_1), (\rho_2,\theta_2), \ldots, (\rho_N,\theta_N)\}$.

We cluster the polar coordinate points belonging to the same wake into one category, so that all wakes can be retrieved, and the position of the line can be accurately located. Here, the angle is used as the criterion to distinguish different wakes. For each two detected lines, if Equation (8) is satisfied, then the two lines are different wakes.

$$|\theta m - \theta n| > \Phi/2 \tag{8}$$

In fact, the wake has a certain width, and a single wake also corresponds to several extreme points, that is, several lines. For each two lines, if Equation (9) is satisfied, we consider that they to belong to the same wake.

$$\begin{cases} |\theta m - \theta n| < \varepsilon \\ |\rho m - \rho n| < \mathrm{w} \end{cases} \tag{9}$$

where, Φ is the angle between the turbulent wake and the Kelvin wake and ε and w are small values which can be set according to the actual situation.

At this point, we can divide the peak point set into different subsets which represent the point clusters corresponding to different wakes and then calculate the center of each point cluster, which is the precise position of each wake, where the center of the N_i-th cluster is:

$$(\overline{\rho}, \overline{\theta})_{Ni} = \sum_{1}^{Ni} (\rho, \theta)/Ni \tag{10}$$

2.3. Wake Scale Measurement

In the following, we measure the visible length of the positioned wake. In reality, the duration of wake formed by ship and its speed determine the length of wake, which can also be affected by external factors such as sea conditions [42]. The process from wake formation to being submerged in the sea clutter is represented in the SAR image as the gradient change of the pixel gray level on the wake line [36].

Measuring the length of the wake is determining the start and end positions of the wake. Unlike sea clutter, the wake area has obvious pixel characteristics. In order to separate the wake pixels from the sea clutter pixels, the statistical pixel characteristics of the sea clutter in the corresponding subwindow are selected as the parameters to distinguish the wake lines from the sea clutter [20].

$$\mu_w = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} f(i,j)}{N \times N} \tag{11}$$

$$\sigma_w = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \left(f(i,j) - \mu_w\right)^2}{N \times N}} \tag{12}$$

where μ_w is the mean gray value of the image, σ_w is the gray standard deviation of the image, and f (i, j) represents the gray value of image pixels.

A set of pixel points g_n is extracted along the axis where the wake is located. Clutter noise existing on the wake axis causes disorganized changes of pixel gradient on the axis. Therefore, this group of one-dimensional data is processed to ensure the smoothness of pixel values on wake lines. The processed data G_n was used to draw the gradient diagram of its wake axis, as shown in Figure 4. The decision rule based on the statistical characteristics of pixel gray scale is:

$$\begin{cases} |Gi - \mu_w| \geq t\sigma_w \\ Dend = 0 \end{cases} \quad , i \in [1, 2, \ldots n] \tag{13}$$

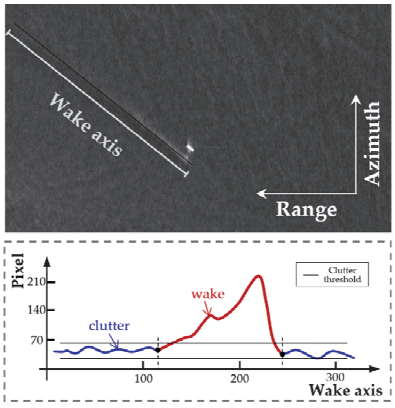


Figure 4. Pixel variation trend diagram of wake axis.

The pixel segment conforming to Equation (13) is considered a wake line. Here, t can be adjusted according to different sea conditions and imaging modes, $D_i = G_{i+1} - G_{i-1}$ represents the gradient of the wake axis and D_{end} is the gradient of the two ends of the wake. When the gradient is zero, the wake is submerged in sea clutter, which is the shortest wake detected.

3. Results

This paper uses the position generated by the specific window search to perform multi-ship and multi-scale wake detection. We will introduce in detail the wake SAR image data set used for the experiment and the execution process of wake detection and analyze the results. The detailed steps of the experiment are as follows, Algorithm 1:

Algorithm 1: Specific Windows Search for Wake Localization and Length Detection

Input: The input is a marine SAR image with ships and their wakes, as well as a variety of other noise.

Process:

1. After preprocessing, obtain the center of the highlighted region $p = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$.
2. Generate a series of bounding boxes $W = \{W_1, W_2, \ldots, W_n\}$ around the center of mass, then calculate the average value of pixels in each box and mask the highlighted pixel area, where the W_i represents the set of windows around the i-th region, which contains windows of different scales and locations.
3. Repeat step 4 for $p = 1, ..., n$.
4. Perform the Radon-based algorithm for each window in the $w^i_{j,k}$. Select the peak points in the Radon domain. Use the clustering algorithm to select the congregate points which are very close. Calculate the gravity centers of the selected clusters.
5. For all labeled locations, measure the wake length.

Output: The output is a set of ship wake line positions with the wake lengths.

3.1. Data Set

Our data set comes from SAR images of offshore China taken by Gaofen-3 satellite, a C-band multi-polarization SAR satellite launched by China in 2016. The orbit parameters and load indexes of Gaofen-3 are shown in Table 1 [35].

Table 1. Orbit parameters and load indexes of Gaofen-3 satellite.

Satellite	Item	Parameters
Gaofen-3	Orbit	Sun-synchronous orbit
	Orbit altitude	755 km
	Orbit inclination	98.5°
	Revisit period	<3 days (Dual-side Looking) <1.5 days (Single-side Looking) [1]
	Frequency band	C-band
	Incidence angle	10°–60°
	Signal bandwidth	0–240 MHz
	Polarization	Single/Dual/Full
	Imaging modes	12
	Spatial resolution	1–500 m
	Swath width	10–650 km

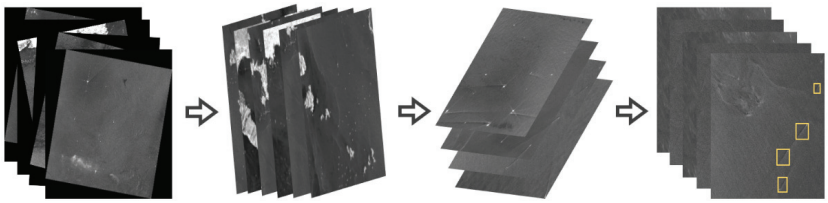
[1] 10 m resolution, 100 km mapping bandwidth, 90% real-time observation area.

The imaging modes of the wake SAR images in the data set were Ultra-Fine Strips (UFS) and Fine Strips (FS) [35]. The detailed information of the data set SAR images is listed in Table 2.

Table 2. The detailed information of the wake data set.

Imaging Mode	Resolution(m)	Incidence Angle (°)
UFS	3 × 3	20–50
FS-I	5 × 5	19–50
FS-II	10 × 10	19–50

In these images, samples containing more than two wakes are manually identified and collected, and then the locations of wakes are marked so as to build the multi-ship and multi-scale data set for algorithm testing, as shown in Figure 5.



(**a**) (**b**) (**c**) (**d**)

Figure 5. Creation of multi-ship and multi-scale wake data sets: (**a**) SAR images; (**b**) preselect the images with ship wakes; (**c**) crop sub-images with multiple ships and multiple wakes; (**d**) labeled sub-images.

3.2. Experimental Results

We selected a 700 × 700 pixel-sized sample with multi-scale and multi-wake in the Gaofen-3 data set, see Figure 6, and the imaging mode was ascending left view. There are three visible wakes in the image, among which Wake #2 is small and difficult to find. The Kelvin wake and turbulent wake in Wake #3 are visible. This image is used as an example to demonstrate the performance of the algorithm.

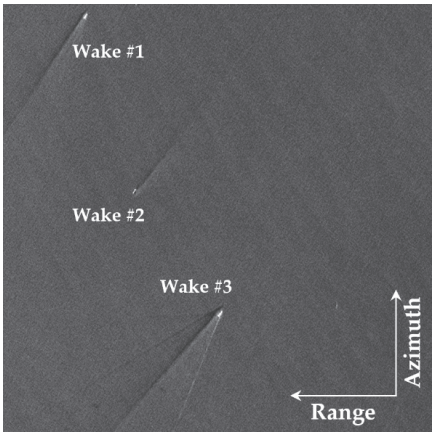


Figure 6. Representative SAR images of ship wakes.

Since we do not need to determine whether the highlights are ship targets at this stage, we use simple morphological processing instead of the complex traditional CFAR algorithm to obtain the highlighted pixels, and the results are shown in Figure 7a. As we can see, although the ship target of Wake #2 is only a few pixels, we can still locate the pixels, which is helpful for the subsequent wake localization.

For these highlighted areas detected, specific windows are generated in their centers and the windows containing the wake are identified. We take Wake #3 with Kelvin wake and turbulent wake as an example for coarse detection and precise location of wake.

In Figure 7b, Radon domain results of the four subwindows show that the maximum score can be obtained at the lower left corner, so it is confirmed that this subwindow contains wake. After standardization, the wake lines corresponding to these preliminary candidate points are shown in Figure 7c. In the stage of fine positioning of each component of the wake, the maximum peak point was taken as the first clustering center [43], and the rest of the clustering centers were determined by geometric and angular relations of the wake; we set ε as 5° and ω as 3 pixels. The results are shown in Figure 7d. It can be shown that the turbulent wake and one Kelvin arm were well detected, while the other Kelvin arm failed to be located due to weak features.

Figure 7e shows the measurement results of wake length. Parameter t is first selected within a reasonable range according to experience, and then manually adjusted and gradually optimized. Here, t is set to 0.35.

Figure 7f is the recognition result of You Only Look Once (YOLO) algorithm [44]. It can be seen that the short wakes have missed detection due to the extremely weak wake characteristics, and the other two wakes were detected with high confidence. It is worth mentioning that, under the condition of no ship target information, for the multi-ship and multi-scale wake detection task, our traditional method can also achieve the results of deep learning methods, and doesn't need a lot of data set as a support. Compared with direct wake line positioning of our algorithm, the wake bounding-box of YOLO requires further line detection in the local area. In short, our algorithm could match the detection effect of the latest deep learning method. (It should be noted that the deep learning method here is based on the standard YOLOv3 network training result. This result is only a preliminary attempt of wake detection using deep learning. We have also conducted new research in the follow-up, and the research results will be announced later.)

In current target detection tasks, the confusion matrix is often used to define some indicators to quantitatively analyze the performance of the algorithm. For ship wake detection, we also define the corresponding 2×2 Confusion Matrix, as shown in Table 3.

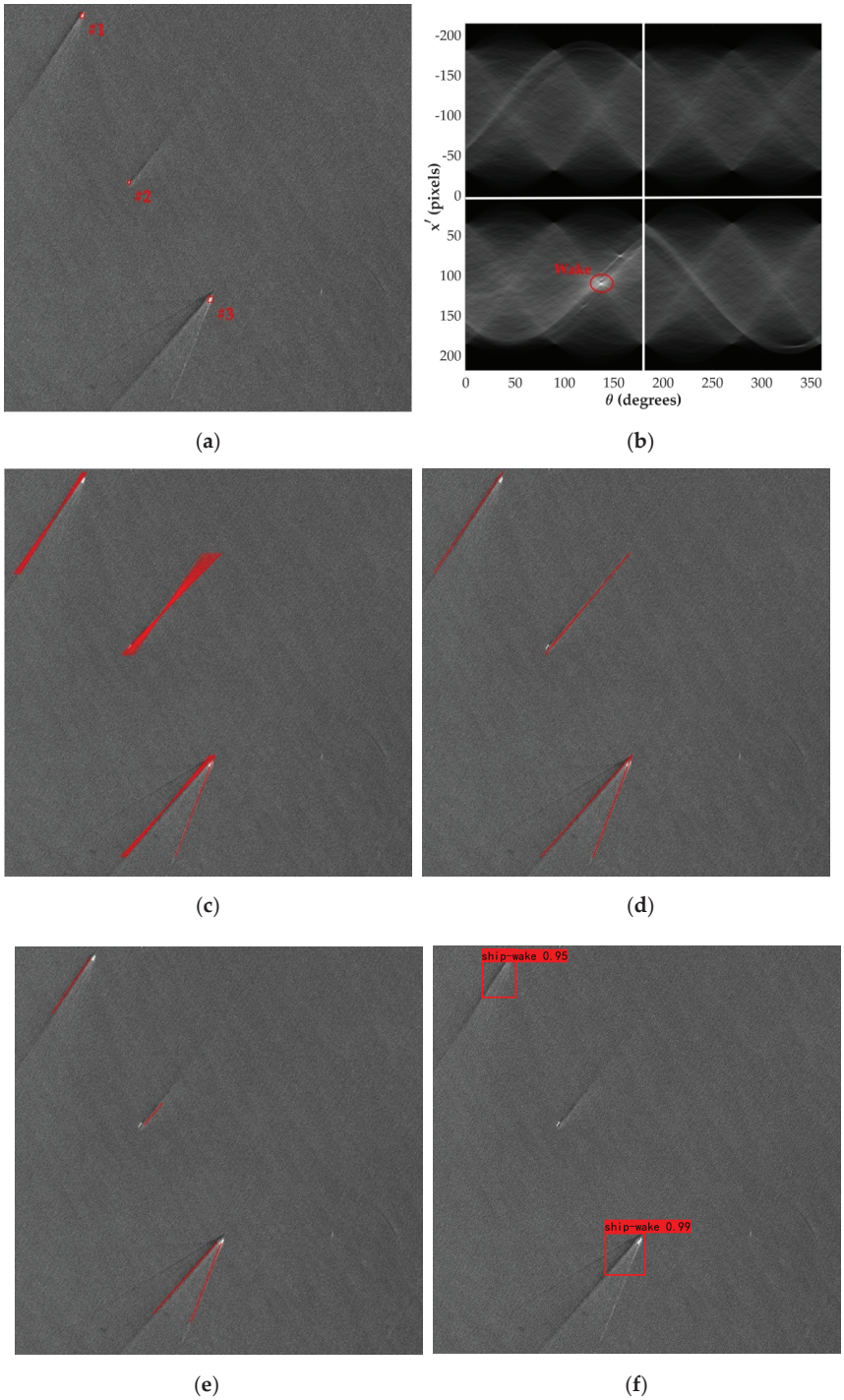


Figure 7. Detection results: (**a**) Highlight area centroid detection results; (**b**) Radon-based results of subwindows with wakes; (**c**) Candidate position of wake line after standardization; (**d**) Fine position of each wake component; (**e**) The final result; (**f**) The YOLO algorithm detection result.

Table 3. Confusion Matrix.

Wake Detection	Prediction = 1	Prediction = 0
Actual = 1	TP	FN
Actual = 0	FP	TN

In the table, TP represents true positive, that is, the detected wake is a true wake; FP stands for a false positive, that is, a fake feature is detected as a wake; FN stands for a false negative, which means that the ship pixel is detected but the wake position is not correctly located; TN stands for true negative, which means that the highlighted pixels of fake-ships are detected and removed. We set the rate of Precision and Recall to evaluate the performance of wake detection:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

In addition, our method can measure the length of visible wake, and Intersection-over-Union (IoU) is a very appropriate indicator to evaluate our detection results and analyze the degree of coincidence between the prediction wake line and the actual wake line. As shown in Figure 8a, the rectangle with the wake line as the diagonal can be considered as its position box, so that the wake detection results can be evaluated in terms of area; IoU can be expressed as:

$$\text{IoU} = \frac{\text{Intersection}}{\text{Union}} \tag{16}$$

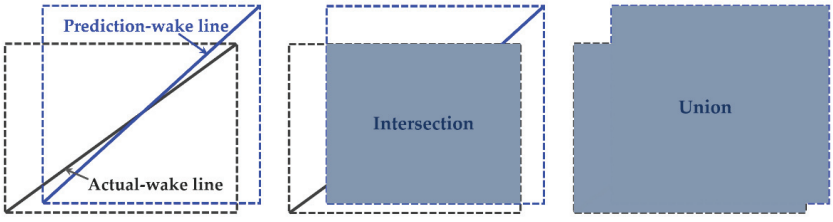


Figure 8. Schematic diagram of IoU principle of wake detection: (**a**) The actual box (Black) and prediction box (Blue) of the wake line; (**b**) Intersection area of actual and prediction boxes; (**c**) Union area of actual and prediction boxes.

That is, the ratio of the intersection area to the union area.

We selected 30 SAR images with multi-ship and multi-scale wakes for the experiment, including 75 visible wakes. The YOLO algorithm was also introduced for comparison, and the experimental results are listed in Table 4.

Table 4. Quantitative comparison results (Shown as average values).

Wake Detection	Our Method	YOLO
Precision	0.91	0.94
Recall	0.89	0.87
IoU	0.82	0.74

From the table data, it can be concluded that, compared with the advanced deep learning algorithm YOLO, the lower Precision of our method is due to some false detection caused by

the lower highlight pixel threshold; the higher Recall indicates that the method can basically not miss the visible wakes or different wake components. In particular, IoU is an evaluation indicator that other traditional methods do not have, and here we achieve even better results than YOLO, which is due to the difficulty of obtaining the pixel-level feature differences of the wake in the feature extraction part of YOLO. Overall, the algorithm can achieve nearly the same level of results as the advanced deep learning algorithm YOLO.

Part of the representative multi-ship and multi-scale ship wake detection results under optimal parameters are shown in Figure 9. It is not difficult to see that Figure 9a is a single dark wake, and this method can accurately locate the wake line and measure the length. Figure 9b–d are all bright wakes with complex backgrounds or speck noise, among which Figure 9b is a short wake with interference of other linear structures. The Pixel-based approach of the algorithm greatly reduces the range of the search area, and the local processing method can effectively avoid the influence of useless areas on the Region of Interest (ROI), achieving good detection results. Figure 9c shows the wake of two ships sailing in a single line, the algorithm can effectively detect the two collinear wakes, rather than just one line running through the whole picture. Figure 9d has multiple wakes with relatively close distances and the results show that mutual interference between the wakes can be avoided. It should be pointed out that there is some highlighted region of the detected fake-ship as shown in Figure 9b,d (the white arrow), which can be removed by the algorithm through the subsequent discrimination principle, and other wakes can be accurately detected. In fact, the result of missed detection is much more serious than false wake alarms, and we try to ensure that all potentially highlighted pixels are detected, even at the cost of extra computation.

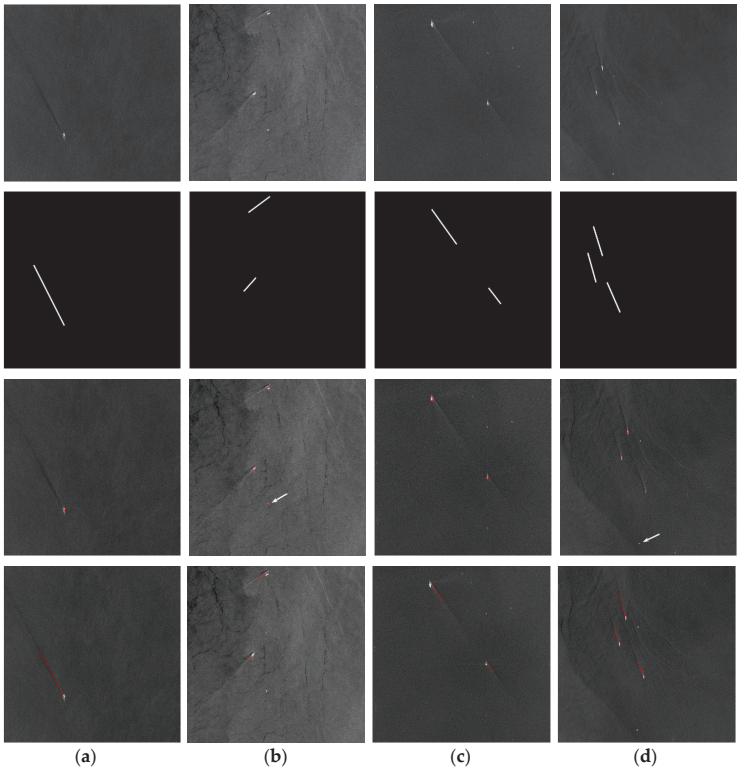


Figure 9. Some SAR samples and detection results of the ship wake: (**a**) Single wake; (**b**–**d**) Multiple wakes.

The above test cases show that there are still missed detections of multi-mode wakes and some small wakes. In order to reduce false detections and missed detections, we still need further improvement in the algorithm. Moreover, the parameter setting of the wake measurement part of the algorithm is conservative, so the statistical characteristics of the wake and sea clutter need to be more deeply explored in the wake measurement part.

4. Discussion

Multi-ship and multi-scale wake detection tests are performed using the collected Gaofen-3 SAR data. Despite not having a priori ship positions, the results show that the method has good capabilities for test samples with multiple wakes. Especially for some local small-sized wakes, which can also be accurately and completely located. In addition, we compared the results of the proposed algorithm and the YOLO algorithm, and, in terms of recognition accuracy, the method almost achieves the effects of the deep learning algorithm without a large amount of training data as the basis. However, some very small-sized ships only exist as a single pixel in the SAR image, and their wakes are tiny and fuzzy. Most algorithms can hardly achieve good results for this type of wake detection. This is also a limitation of this algorithm because it is difficult to determine whether these bright spots are ships or speckle noise.

In fact, our specific window search method provides a new solution in which ships and wakes are detected as a partner instead of being detected individually and unrelatedly. The performance of the algorithm for multi-ship and multi-scale wake detection can be intuitively displayed in both the Radon domain and image domain. Future research work should focus on the detection of very small wakes and the detection of wakes in more complex environments. We need to further improve our detection logic or solve speckle noise suppression more deeply.

5. Conclusions

This paper proposes a specific window search method for rapid detection of multi-ship and multi-scale wakes in SAR images. We have observed that in SAR images, such as the Gaofen-3 offshore China data set we use, there are often multiple wakes of different positions and lengths within a certain range, and the wake targets always appear as line features occupying a small area. Therefore, the single wake extraction algorithm can never capture all possible ship wake positions. Aiming at the problem of multi-ship and multi-scale wake detection, we introduce a specific window search method, which is different from most pre-selection box generation methods for target detection, and is also different from the sliding window style global scan search. Considering the strong geometric correlation between ships and wakes and their typicality in pixels, we generate a specific search sub-window based on the highlighted pixel area in the image, so that the area that needs to be detected is greatly reduced. Through the localized Radon-based enhancement algorithm, the real ship target area can be screened out, and the sub-window that is bounding the wake can be determined. Subsequently, combined with the geometric angle relationship, our method can accurately locate the wake axis, capture the different components of the wake and then cluster the candidate wakes point clusters so as to reconstruct all the wakes. Finally, through empirical analysis of multiple samples, and based on pixel statistics, the shortest visible length of the wake can be measured.

Author Contributions: Conceptualization, K.D. and J.Y.; methodology, K.D.; software, K.D. and J.Y.; validation, K.D.; formal analysis, K.D. and Z.W.; investigation, Z.W.; resources, K.D. and Q.Z.; writing—original draft preparation, K.D.; writing—review and editing, K.D., Q.Z., J.Y. and K.N.; visualization, K.D.; supervision, Q.Z., X.W. and K.N.; project administration, Q.Z. and K.N.; funding acquisition, Q.Z. and K.N. All authors have read and agreed to the published version of the manuscript.

Funding: Shenzhen Fundamental Research Funding (JCYJ20200109143008165, JCYJ20210324115813037).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from the National Center for Satellite Marine Applications and are available https://osdds.nsoas.org.cn (accessed on 16 May 2021) with the permission of the National Center for Satellite Marine Applications.

Acknowledgments: The GF—3 satellite data acquisition from the website: https://osdds.nsoas.org.cn (accessed on 16 May 2021) The authors thank the National Center for Satellite Marine Applications for their data support.

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

References

1. Xu, P.; Li, Q.; Zhang, B.; Wu, F.; Zhao, K.; Du, X.; Yang, C.; Zhong, R. On-Board Real-Time Ship Detection in HISEA-1 SAR Images Based on CFAR and Lightweight Deep Learning. *Remote Sens.* **2021**, *13*, 1995. [CrossRef]
2. Ma, M.; Chen, J.; Liu, W.; Yang, W. Ship classification and detection based on CNN using GF-3 SAR images. *Remote Sens.* **2018**, *10*, 2043. [CrossRef]
3. Yekeen, S.; Balogun, A.; Yusof, K. A novel deep learning instance segmentation model for automated marine oil spill detection. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 190–200. [CrossRef]
4. Kang, M.; Baek, J. SAR Image Change Detection via Multiple-Window Processing with Structural Similarity. *Sensors* **2021**, *21*, 6645. [CrossRef]
5. Gao, Y.; Gao, F.; Dong, J.; Wang, S. Change Detection from Synthetic Aperture Radar Images Based on Channel Weighting-Based Deep Cascade Network. *IEEE J. Sel. Top. Appl. Earth Observ.* **2019**, *12*, 4517–4529. [CrossRef]
6. Zhang, X.; Liu, G.; Zhang, C.; Atkinson, P.M.; Tan, X.; Jian, X.; Zhou, X.; Li, Y. Two-phase object-based deep learning for multi-temporal SAR image change detection. *Remote Sens.* **2020**, *12*, 548. [CrossRef]
7. Chen, H.; Shi, Z. A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection. *Remote Sens.* **2020**, *12*, 1662. [CrossRef]
8. Niedermeier, A.; Romaneessen, E.; Lehner, S. Detection of coastlines in sar images using wavelet methods. *IEEE Trans. Geosci. Remote Sens.* **2000**, *38*, 2270–2281. [CrossRef]
9. Baselice, F.; Ferraioli, G. Unsupervised coastal line extraction from sar images. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1350–1354. [CrossRef]
10. Ouchi, K. Recent trend and advance of synthetic aperture radar with selected topics. *Remote Sens.* **2013**, *5*, 716–807. [CrossRef]
11. Reed, A.M.; Milgram, J.H. Ship Wakes and Their Radar Images. *Annu. Rev. Fluid Mech.* **2002**, *34*, 469–502. [CrossRef]
12. Touzi, R.; Charbonneau, F.J.; Hawkins, R.K.; Vachon, P.W. Ship detection and characterization using polarimetric SAR. *Can. J. Remote Sens.* **2004**, *30*, 552–559. [CrossRef]
13. Panico, A.; Graziano, M.D.; Renga, A. SAR-Based Vessel Velocity Estimation from Partially Imaged Kelvin Pattern. *IEEE Trans. Geosci. Remote Sens.* **2017**, *14*, 2067–2071. [CrossRef]
14. Graziano, M.D.; Rufino, G.; D'Errico, M. Wake-based ship route estimation in high-resolution SAR images. *Proc. SPIE Int. Soc. Opt. Eng.* **2014**, *9243*. [CrossRef]
15. Wang, J.; Ci, L. Ship's Length Estimation from Its Wakes in Synthetic Aperture Radar Images. *Trans. Beijing Inst. Technol.* **2004**, *24*, 901–904.
16. Rey, M.T.; Tunaley, J.K.; Folinsbee, J.T.; Jahans, P.A.; Dixon, J.A.; Vant, M.R. Application Of Radon Transform Techniques To Wake Detection In Seasat-A SAR Images. *IEEE Trans. Geosci. Remote Sens.* **1990**, *28*, 553–560. [CrossRef]
17. Skingley, J.; Rye, A.J. The Hough transform applied to SAR images for thin line detection. *Pattern Recognit. Lett.* **1987**, *6*, 61–67. [CrossRef]
18. Kang, K.; Kim, D. Ship Velocity Estimation From Ship Wakes Detected Using Convolutional Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 4379–4388. [CrossRef]
19. Zilman, G.; Zapolski, A.; Marom, M. The speed and beam of a ship from its wake's SAR images. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 2335–2343. [CrossRef]
20. An, Q.; Pan, Z.; You, H. Ship detection in Gaofen-3 SAR images based on sea clutter distribution analysis and deep convolutional neural network. *Sensors* **2018**, *18*, 334. [CrossRef] [PubMed]
21. Jin, M.K.; Chen, K.S. The application of wavelets correlator for ship wake detection in SAR images. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1506–1511. [CrossRef]
22. Courmontagne, P. An improvement of ship wake detection based on the radon transform. *Signal Process.* **2005**, *85*, 1634–1654. [CrossRef]
23. Arnold-Bos, A.; Martin, A.; Khenchaf, A. Obtaining A Ships Speed and Direction from Its Kelvin Wake Spectrum Using Stochastic Matched Filtering. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Barcelona, Spain, 23–28 July 2007; pp. 1106–1109.

24. Biondi, F. A Polarimetric Extension of Low-Rank Plus Sparse Decomposition and Radon Transform for Ship Wake Detection in Synthetic Aperture Radar Images. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 75–79. [CrossRef]
25. Yang, G.; Jing, Y.; Xiao, C.; Sun, W. Ship wake detection for SAR images with complex backgrounds based on morphological dictionary learning. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, China, 20–25 March 2016; pp. 1896–1900.
26. Copeland, A.C.; Ravichandran, G.; Trivedi, M.M. Localized Radon transform-based detection of ship wakes in SAR images. *IEEE Trans. Geosci. Remote Sens.* **1995**, *33*, 35–45. [CrossRef]
27. Ai, J.; Qi, X.; Yu, W.; Deng, Y.; Liu, F.; Shi, L.; Jia, Y. A Novel Ship Wake CFAR Detection Algorithm Based on SCR Enhancement and Normalized Hough Transform. *IEEE Geosci. Remote Sens. Lett.* **2011**, *8*, 681–685.
28. Du, G.; Yeo, T.S. A novel Radon transform-based method for ship wake detection. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Anchorage, AK, USA, 20–24 September 2004; pp. 3069–3072.
29. Cusano, M.; Lichtenegger, J.; Lombardo, P.; Petrocchi, A.; Zanovello, D. A real time operational scheme for ship traffic monitoring using quick look ERS SAR images. In Proceedings of the IEEE International Geoscience & Remote Sensing Symposium, Honolulu, HI, USA, 24–28 July 2000; pp. 2918–2920.
30. Yang, G.; Yu, J.; Sun, W. Ship wake detection in SAR images with complex backgrounds based on relative total variation. *J. Univ. Chin. Acad. Sci.* **2017**, *34*, 734–742.
31. Graziano, M.D. Preliminary Results of Ship Detection Technique by Wake Pattern Recognition in SAR Images. *Remote Sens.* **2020**, *12*, 2869. [CrossRef]
32. Graziano, M.D.; Renga, A. Towards Automatic Recognition of Wakes Generated by Dark Vessels in Sentinel-1 Images. *Remote Sens.* **2021**, *13*, 1955. [CrossRef]
33. Tings, B.; Pleskachevsky, A.; Velotto, D.; Jacobsen, S. Extension of ship wake detectability model for non-linear influences of parameters using satellite based x-band synthetic aperture radar. *Remote Sens.* **2019**, *11*, 563. [CrossRef]
34. Tings, B.; Velotto, D. Comparison of ship wake detectability on C-band and X-band SAR. *Int. J. Remote Sens.* **2018**, *39*, 4451–4468. [CrossRef]
35. China Ocean Satellite Data Service Center. Available online: https://osdds.nsoas.org.cn (accessed on 16 May 2021).
36. Sun, Y.; Peng, L.; Jin, Y. Ship Wake Components: Isolation, Reconstruction, and Characteristics Analysis in Spectral, Spatial, and TerraSAR-X Image Domains. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4209–4224. [CrossRef]
37. Fan, K.; Zhang, H.; Liang, J.; Chen, P.; Xu, B.; Zhang, M. Analysis of ship wake features and extraction of ship motion parameters from SAR images in the Yellow Sea. *Front. Earth Sci.* **2019**, *13*, 588–595. [CrossRef]
38. Chen, P.; Li, X.; Zheng, G.; Zhang, H. A new method for extracting ship motion parameters in Radarsat-2 SAR imagery. *Int. J. Remote Sens.* **2019**, *40*, 5617–5634. [CrossRef]
39. Hennings, I.; Romeiser, R.; Alpers, W.; Viola, A. Radar imaging of Kelvin arms of ship wakes. *Int. J. Remote Sens.* **1999**, *20*, 2519–2543. [CrossRef]
40. Jen, K. Theory of synthetic aperture radar imaging of a moving target. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 1984–1992.
41. Yang, J.; Zhang, Y. Analysis on the azimuth shift of a moving target in SAR image. *Prog. Electromagn. Res.* **2015**, *42*, 121–134. [CrossRef]
42. Zilman, G.; Zapolski, A.; Marom, M. On Detectability of a Ship's Kelvin Wake in Simulated SAR Images of Rough Sea Surface. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 609–619. [CrossRef]
43. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496. [CrossRef]
44. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.

electronics

MDPI

Article

An Exploratory Verification Method for Validation of Sea Surface Radiance of HY-1C Satellite UVI Payload Based on SOA Algorithm

Lei Li [1,2,3], Dayi Yin [1,2,3,*], Qingling Li [1,2], Quan Zhang [1,2] and Zhihua Mao [4]

[1] Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai 200083, China
[2] Key Laboratory of Infrared System Detection and Imaging Technology, Chinese Academy of Sciences, Shanghai 200083, China
[3] University of Chinese Academy of Sciences, Beijing 100049, China
[4] States Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of Oceanography, Ministry of Natural Resources, Hangzhou 310012, China
* Correspondence: yindayi@mail.sitp.ac.cn

Abstract: To support the application of ocean surface radiance data from the ultraviolet imager (UVI) payload of the HY-1C oceanographic satellite and to improve the quantification level of ocean observation technology, the authenticity check study of ocean surface radiance data from the UVI payload was conducted to provide a basis for the quantification application of data products. The UVI load makes up for the lack of detection capabilities of modern ocean remote sensing satellites in the ultraviolet band. The UVDRAMS (Ultra-Violet Dual-band RadiAnce Measurement System) was used to verify the surface radiance data collected at 16 stations in the study area and the pupil radiance data collected by the UVI payload to establish an effective radiative transfer model and to identify the model parameters using the seeker optimization algorithm (SOA). The study of the UVDRAMS measurement system based on the SOA algorithm and the validation of the sea surface radiance of the UVI payload of the HY-1C satellite shows that 97.2% of the incident pupil radiance of the UVI payload is contributed by the atmospheric reflected radiance, and only 2.8% is from the real radiation of the water surface, while the high signal-to-noise ratio of the UVI payload of the HY-1C ocean satellite can effectively distinguish the reflectance of the water body. The high signal-to-noise ratio of the UVI payload of the HY-1C ocean satellite can effectively distinguish the amount of standard deviation in the on-satellite radiation variation, which meets the observation requirements and provides a new way of thinking and technology for further quantitative research in the future.

Keywords: HY-1C; sea surface radiance; SOA algorithm; synchrotron radiation transfer model; UVI; validation

Citation: Li, L.; Yin, D.; Li, Q.; Zhang, Q.; Mao, Z. An Exploratory Verification Method for Validation of Sea Surface Radiance of HY-1C Satellite UVI Payload Based on SOA Algorithm. *Electronics* **2023**, *12*, 2766. https://doi.org/10.3390/electronics12132766

Academic Editor: Federico Alimenti

Received: 17 May 2023
Revised: 15 June 2023
Accepted: 19 June 2023
Published: 21 June 2023

1. Introduction

The ocean plays an important role in global climate and weather change, and marine remote sensing (RS) satellites can be used for effective observation of the marine environment and climate [1]. The continuous development of marine RS satellites enables them to play an increasingly important role in marine disaster prevention and reduction, environmental protection, marine ecology, marine rights protection, resource development, and many other fields [2–4]. The acquisition of marine RS data is a complicated process that is affected by many factors, such as atmospheric radiation transmission characteristics, the RS operating environment, the RS working state, the state of the observed target, etc. To improve the quantification level of marine observation technology and to judge whether the RS data received by the marine RS sensor meet the design requirements and whether the RS inversion products can accurately and truly reflect the actual situation, the validation method of the marine satellite payload needs to be studied [5].

The quantitative application of RS data is an important issue that needs to be solved for the further development of RS technology [6]. The development of quantitative RS technology calls for higher requirements for determining the accuracy of satellite RS data and the quality of products. This requires not only the continuous improvement and development of new RS devices to improve the accuracy of quantitative RS but also accurate calibration of the radiation measurement results of RS devices and checking whether the products of RS data accurately reflect the geophysical parameters detected. Therefore, the validation of RS data is put forward as necessary to study. At present, there are two main methods of validation. One is a direct test [7], that is, to obtain the true value of the ground via synchronous ground measurements with a satellite-borne remote sensor and to compare and analyze the data results with RS data. The other is the indirect test method [8], including cross-validation, which uses other verified satellite products of known accuracy to test satellite products, space–time analysis, the process model, etc.

Due to the serious air pollution problems in some Asian countries, S. V. V. Arun Kumar et al. [9] cross-verified the distribution data on chlorophyll A in the Arabian Sea with SeaWiFS, MODIS-Aqua, MODIS-Terra, and MERIS, and analyzed the differences in the data in different sea areas. In recent years, scientists have also carried out much verification work on RS data from the payload of the GOES-16 geostationary orbit weather satellite, which was launched in 2016. Bartlett B et al. [10,11] used a novel geospatial database and image abstraction techniques to conduct a detector-level in-depth analysis of data from target sites on the Advanced Baseline Imager (ABI) of GOES-16 to provide independent verification of the SI traceability of its spectral radiation observations. Additionally, they established a new performance benchmark for NOAA's next-generation geostationary observation instrument products. In 2016, ESA launched the Sentinel-3A/B satellite, which is equipped with the OLCI (Ocean Land Color Instrument) and SLSTR (Sea and Land Surface Temperature Radiometer) to measure sea temperature, sea color, sea level height, and sea ice thickness. The measured data can be used to monitor the Earth's climate change, marine pollution, and biological productivity. Jungang Yang et al. [12] verified the accuracy and long-term stability of Sentinel-3A SWH by double cross-verifying Sentinel-3A SWH data with NDBC buoy data and Sentinel-3A SWH data with Jason-3 data. In December 2017, Japan's newest generation of the Earth Environment Change Observation Satellite (GCOM) was equipped with a multi-wavelength optical radiometer (SGLI) [13], which has a central wavelength of 380 nm and a bandwidth of 10 nm on the ultraviolet spectrum. After its successful launch, the in-orbit test was conducted [14–16].

In the processing of satellite data, many novel algorithms have also been introduced. Tian, H et al. [17], used the optical image data of Landsat-7 and -8 and Sentinel-2 optical images and used the decision tree classification method to classify winter crops at the pixel level. The overall classification accuracy rate reached 96.22%, making a significant contribution to the rapid and accurate mapping of winter crops. Tian, H. et al. [18] used Sentinel-2, Landsat-8, and Sentinel-1 RS image data to distinguish garlic from winter wheat. Through the cross-coupling of these three satellite data sets to carry out classification extraction, the results show that, compared with single satellite data, the mixed processing of multi-source satellite RS data significantly improves classification accuracy. Anahita Modabberi et al. [19] used the MODIS-Aqua Chl-a data from 2003 to 2017 in the study of eutrophication in the Caspian Sea and innovatively introduced the pod algorithm to extract the dominant features. The research showed that the degradation of the Caspian Sea was significantly accelerated.

These satellite payloads have certain limitations in the observation of ocean RS. Due to the limitation of the observation spectrum of the detectors, the working bands of these satellite payloads are basically distributed in visible light, near-infrared, and other similar bands and lack the ability to observe the optical characteristics of ocean water bodies in the ultraviolet band. Only the SGLI payload carried by the GCOM launched by Japan has observation capabilities in the near-ultraviolet and 380 nm bands. On 7 September 2018, China launched a new generation of ocean RS satellites (HY-1C) [20]. The load of the

ultraviolet imager (UVI) carried on the HY-1C ocean satellite uses a large-field combined ultraviolet transmission optical system and an ultraviolet GaN focal plane detector. It has a resolution of 500 m, a huge width of 2900 km, a high signal-to-noise ratio of 1000 times, and a dynamic response that is 1.2 times the solar dual width. It expands the spectrum coverage of satellites; has an ultraviolet dual band (345 nm~365 nm and 375 nm~395 nm) [21,22]; improves the capabilities of atmospheric correction, CDOM, and carbon cycle monitoring; and provides new means of detecting offshore oil spouts. It is the first time China has used ultraviolet technology to carry out space and marine civilian RS applications.

Limited by the spatial resolution and spectral differences, this also creates another problem. The UVI payload carried on HY-1C cannot be indirectly cross-validated with the observation data of other satellite payloads. To evaluate the authenticity and accuracy of the RS data of ocean surface radiance under a UVI load and the degree of how well the sensor design index meets the requirements, a direct authenticity test was conducted on the ocean surface radiance RS data of the UVI load. In this paper, the ocean surface radiance data was collected by the Ultraviolet Dual-band Radiance Measurement System (UVDRAMS) in September 2018 from 16 stations in the study area, where the main performance and parameters of the UVDRAMS were consistent with the UVI load, and the entrance pupil radiance data were collected by the UVI load. The satellite–ground synchrotron radiation verification was carried out, the satellite–ground synchrotron radiation transmission model was established, and the model parameters were identified using the seeker optimization algorithm (SOA) [23]. According to the established satellite-to-ground synchrotron radiation transfer model, the contribution components of the entrance pupil radiance of the UVI load were analyzed, and it was judged whether the signal-to-noise ratio index of the HY-1C ocean satellite's UVI load can meet the observation requirements to provide further information for the future. Carrying out this quantitative research has laid a technical foundation.

In the study of this paper, the UVI load makes up for the lack of detection capabilities of modern ocean RS satellites in the ultraviolet band. Through the ocean in situ synchronous observation experiment combined with the SOA algorithm, a set of exploratory satellite-to-earth synchrotron radiation transmission models is established, and through this model, the authenticity of the UVI load data is checked.

2. The Validation Method of the HY-1C Satellite's Ultraviolet Imager

2.1. The Validation Method Principle

An authenticity check is an independent method to obtain the reference data representing the ground truth value and to realize the accuracy verification and uncertainty evaluation of RS data or products through comparing and analyzing RS data or products. Broadly speaking, the authenticity check includes checking the authenticity of satellite loads, RS common products, and RS application products [24].

After converting the digital quantities output from the remote sensors into radiometric quantities by calibration, the required physical parameters must be extracted from these radiometric quantities [25]. To determine whether the geophysical information obtained from satellite data correctly reflects the objective existence, i.e., the quality of the satellite data product, it must be evaluated using an independent method, such as performing a veracity test [26,27]. The flow chart of the validation method principle used in this paper is shown in Figure 1.

Satellite products and field-measured data have different temporal-spatial sampling characteristics, and it is necessary to determine a reasonable temporal-spatial window according to the spatial resolution of satellite products, as well as the temporal-spatial variation and uniformity of water bodies, and calculate the mean value of effective pixels in the spatial window and the mean value of pixels in the time window. The mean value of the effective on-site measurement is used as a matching data pair and included in the verification data set. In this experiment, we use the time window of the HY-1C satellite transit to conduct an ocean on-site observation experiment to obtain the ocean surface

ultraviolet radiance L_{sfc}, and at the same time, collect this UVI load entrance pupil radiance data L_{sat} within a period of time. According to the principle of atmospheric radiative transfer, an atmospheric radiative transfer model is constructed.

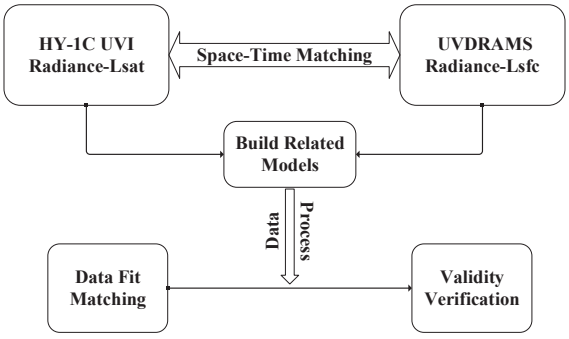


Figure 1. Flowchart of the principle of obtaining and validating the sea surface radiance.

2.2. Modeling of Satellite–Ground Synchrotron Radiation Transport

In the ultraviolet spectral range, it is known from radiative transfer theory that the surface is assumed to be a Lambertian surface, that the downward atmospheric thermal radiation is isotropic, and that the spectral radiation received by the satellite is the total contribution of the interaction between the solar spectral radiation, the atmosphere, and the terrestrial target [28]. The first component is the thermal radiation emitted by the object target, the magnitude of which is determined by the emissivity of the object's surface and the atmospheric transmittance between the target and the satellite; the second component is the reflected radiation from the object target to the total radiation of the downgradient atmospheric radiation, the ambient background radiation, and the thermal radiation component of the solar incidence, which is normally neglected; and the third component is the atmospheric uplink radiation between the object and the satellite, which is related to the content and physical state of the absorbing gas in the atmosphere. In the case of ocean observations, the observational model is simplified, and the radiance observed in space by the UVI payload is shown in Equation (1) [29].

$$L_{sat} = t \times L_{sfc} + L_{sky_{top}} = \frac{L_{sky\_trans}}{L_{sky}} \times L_{sfc} + L_{sky\_ref} \quad (1)$$

where L_{sat} is the radiance received by the pupil of the satellite load sensor; t is the total atmospheric transmittance, which is determined by the skylight upward radiation L_{sky} and the skylight upward radiation through the atmosphere L_{sky_trans}; L_{sfc} is the in situ measured radiance from the sea surface upwards; and L_{sky_ref} is the atmospheric reflected radiance.

Assuming that the nature of the thermal radiation L_{sfc} is uniform in waters of a similar sea state in the ocean, then by accurately measuring the surface upward radiance and performing simultaneous verification analysis with the incoming pupil radiance L_{sat} collected by the UVI load, t and L_{sky_ref} can be obtained. For the determination of the pupil radiance L_{sat} of the UVI payload, a look-up table of radiance and sensor DN values was established by integrating sphere radiometric calibration before the satellite launch, and the pupil radiance L_{sat} can be obtained from the corresponding DN values of the UVI payload. The sea surface ultraviolet radiance measured synchronously on site is the basis for the verification of satellite–terrestrial synchrotron radiation, and for in situ optical measurements of ocean waters, in situ observations using the above-water method [30] can yield the sea surface upward radiance L_{sfc} and the skylight upward radiation L_{sky} of the UV band. The total atmospheric transmittance t and atmospheric reflectance L_{sky_ref} are obtained by fitting the ocean radiance data from multiple regional stations into Equation (1).

The synchrotron radiation transport model was converted to form a one-dimensional linear regression equation, as shown in Equation (2).

$$y = ax + b \tag{2}$$

where y represents L_{sat}; a represents the total atmospheric transmittance; x is the in situ measured radiance L_{sfc} from the sea surface upwards; and b is the atmospheric reflected radiance L_{sky_ref}. The in situ observation data from some stations are selected and combined with the satellite-based simultaneous RS data, and the optimal solutions for a and b can be obtained by fitting and optimizing with the intelligent optimization algorithm.

2.3. Synchrotron Radiation Transmission Model Parameter Identification Method Based on SOA Algorithm

The seeker optimization algorithm (SOA) simulates the random search behavior of humans and directly applies the intelligent search behavior of humans to the search for optimization problem solutions [31]. In optimization calculations, human random search behavior can be understood in this way: in the search process of continuous space, there may be a better solution around the solution, and the optimal solution may exist in the neighborhood of the better solution. Therefore, when the searcher is in a better position, they should search in a smaller neighborhood. When the searcher is in a poor position, they should search in a larger neighborhood [32,33]. To this end, the SOA uses fuzzy logic that effectively describes the natural language and uncertain reasoning to model the above search rules and determine the search step size. The SOA obtains social experience and cognitive experience through social learning and cognitive learning, respectively, and determines the direction of the individual search by combining the self-organizing aggregation behavior of intelligent groups, self-centered egoistic behavior, and human pre-action behavior [34].

The uncertain reasoning behavior of the SOA uses the approximation ability of the fuzzy system to simulate human intelligent search behavior and to establish the connection between perception (objective function value) and behavior (step size) [35]. The Gaussian membership function is used to represent the fuzzy variable of the search step size, as shown in Equation (3):

$$u_A(x) = \exp\left[-(x-u)^2/2\delta^2\right] \tag{3}$$

where u_A is the Gaussian membership degree, x is the input variable, and u and δ are membership function parameters.

The fuzzy variable "small" of the objective function adopts a linear membership function so that the membership degree is directly proportional to the order of the function values, i.e., the maximum membership value $u_{\max} = 1.0$ in the best position, the minimum membership value $u_{\min} = 0.0111$ in the worst position, and so on for other positions. This can be expressed by Equations (4) and (5):

$$u_i = u_{\max} - \frac{s - I_i}{s - I}(u_{\max} - u_{\min}), i = 1, 2, \cdots, s \tag{4}$$

$$u_{ij} = rand(u_i, 1), j = 1, 2, \cdots, D \tag{5}$$

where u_i is the membership degree of the objective function value i; u_{ij} is the membership degree of the objective function value i in the j-dimension search space; I_i is the sequence number of $x_i(t)$ after the population function values are arranged in descending order; and D is the dimension of the search space.

After obtaining the membership degree u_{ij}, the step size can be obtained according to the behavior of uncertain reasoning, as shown in Equation (6):

$$\alpha_{ij} = \delta_{ij}\sqrt{-\ln(u_{ij})} \tag{6}$$

where α_{ij} is the search step size of the j-dimension search space, and δ_{ij} is the parameter of the Gaussian membership function, whose value can be determined by Equations (7) and (8):

$$\vec{\delta_{ij}} = \omega \cdot abs\left(\vec{x}_{\min} - \vec{x}_{\max}\right) \tag{7}$$

$$\omega = (T_{\max} - t)/T_{\max} \tag{8}$$

where $x_{\min}$ and $x_{\max}$ are the positions in the same subgroup with the minimum and maximum functional values, respectively; ω is the inertia weight, which decreases linearly from 0.9 to 0.1 with the increase in evolutionary algebra; t and $T_{\max}$ are the current iteration number and the maximum iteration number, respectively; and the function $abs(\cdot)$ takes the absolute value of each entry.

Through the analysis and modeling of human egoistic behavior, altruistic behavior, and pre-acting behavior, the egoistic direction $\vec{d}_{i,ego}$, altruistic direction $\vec{d}_{i,alt}$, and pre-acting direction $\vec{d}_{i,pro}$ of any i-th search individual are obtained, respectively, as shown in Equations (9)–(11):

$$\vec{d}_{i,ego}(t) = \vec{p}_{i,best} - \vec{x}_i(t) \tag{9}$$

$$\vec{d}_{i,alt}(t) = \vec{g}_{i,best} - \vec{x}_i(t) \tag{10}$$

$$\vec{d}_{i,pro}(t) = x_i(t_1) - x_i(t_2) \tag{11}$$

The searcher considers all factors and determines the search direction by using a randomly weighted geometric average of the three directions, as shown in Equation (12):

$$\vec{d}_i(t) = sign\left(\omega \vec{d}_{i,pro} + \varphi_1 \vec{d}_{i,ego} + \varphi_2 \vec{d}_{i,alt}\right) \tag{12}$$

where $\vec{x}_i(t_1)$ and $\vec{x}_i(t_2)$ are the best positions in $\left\{\vec{x}_i(t-2), \vec{x}_i(t-1), \vec{x}_i(t)\right\}$; $g_{i,best}$ is the best position based on the collective history of the neighborhood where the i-th search individual is located and $p_{i,best}$ is the best position that the i-th search individual has experienced thus far; $sign(\cdot)$ denotes the sign function of each dimension of the input vector; φ_1 and φ_2 are real numbers uniformly and randomly selected in the known interval [0, 1]; and ω is the inertia weight, which decreases linearly from 0.9 to 0.1 with the increase in evolutionary algebra.

After determining the search direction and step size, the position is updated, as shown in Equations (13) and (14):

$$\Delta x_{ij}(t+1) = \alpha_{ij}(t) d_{ij}(t) \tag{13}$$

$$x_{ij}(t+1) = x_{ij}(t) + \Delta x_{ij}(t+1) \tag{14}$$

3. Marine In Situ Observation Field Test Verification

The Ultraviolet Dual-band Radiance Measurement System (UVDRAMS), developed by the Shanghai Institute of Technical Physics, Chinese Academy of Sciences, is used in this marine observation experiment. Its main performance and parameters are consistent with the UVI load. The UVDRAMS has an ultraviolet dual band (345 nm~365 nm and 375 nm~395 nm), high sensitivity, and a large dynamic range. It has two dynamic ranges, covering 0.4~0.5 times the solar constant and up to 1.2 times the solar constant. The two dynamic ranges can simultaneously obtain high signal-to-noise ratio observation data. The performance parameters are shown in Table 1.

Table 1. The specifications of UVDRAMS.

Ultraviolet Dual-Band Radiance Measurement System		
Detector spectrum	B1 (345 nm~365 nm)	B2 (375 nm~395 nm)
Center wavelength	355 nm	385 nm
SNR	>1000 (typical radiance of 7.5 mW·cm^{-2}·um^{-1}·sr^{-1})	>1000 (typical radiance of 6.1 mW·cm^{-2}·um^{-1}·sr^{-1})
Dynamic range (mW·cm^{-2}·um^{-1}·sr^{-1})	High dynamic of 35.6 Low dynamic of 17.5	High Dynamic of 36.1 Low Dynamic of 18.6
FOV	23°	
Angular resolution	0.68 mrad	
Absolute radiometric calibration accuracy	<5%	

The module structure of the UVDRAMS is shown in Figure 2.

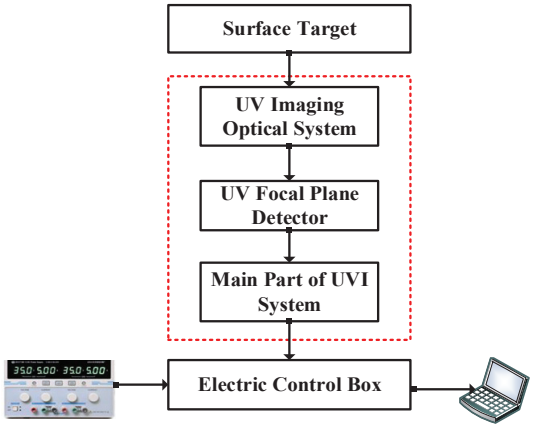


Figure 2. The module structure of UVDRAMS.

The schematic diagram of the UVDRAMS marine observation process [36] is shown in Figure 3.

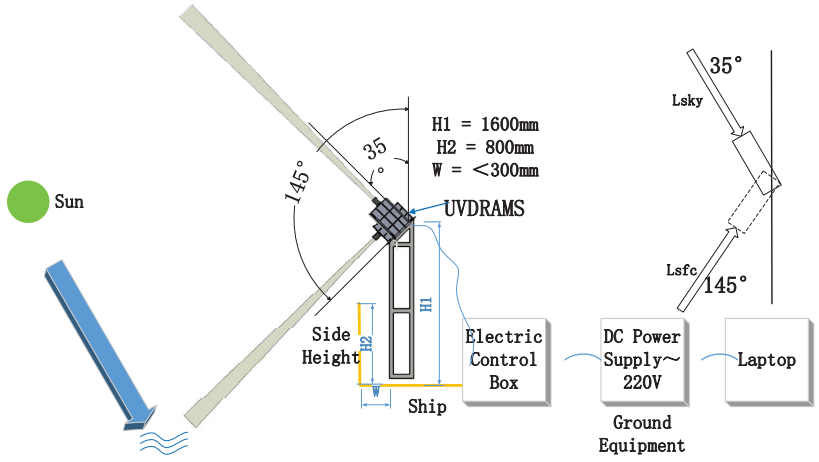


Figure 3. Schematic diagram of sea surface observation.

To ensure the validity and stability of the observation results, the sea area of the observed experiment needs to be relatively stable in space and time [37–40]. The main site

of the observation experiment is located in the northern South China Sea (E107.5°~113.5°, N16°~20°) around Hainan Island, China. It covers coastal turbid water and offshore clean water with a maximum depth of more than 1000 m. In situ sea surface radiation measurements were carried out from 12 September to 14 October 2018 from Zhoushan Island, Zhejiang, China, to the test site in the northern South China Sea. The observation time point was selected to be 0.5 h before and after the transit of HY-1C, and the time correlation of the observation data was maintained. To increase the number of simultaneous observation tests and obtain more observation data, observation tests were also carried out along the route. A total of 17 observation tests were carried out throughout the voyage when the weather and sea conditions allowed. There were 3 stations in the East China Sea area and 13 stations in the South China Sea area around Hainan Island. Specific test site statistics and experimental sea areas are shown in Table 2 and Figure 4.

Table 2. The experiment results at each station.

Study Area	Geographical Location (Longitude, Latitude)	L_{sfc} (mW·cm^{-2}·um^{-1}·sr^{-1})		L_{sky} (mW·cm^{-2}·um^{-1}·sr^{-1})	
		Average Radiance	Standard Deviation	Average Radiance	Standard Deviation
DH01	N30.40.36 E122.53.91	0.774	0.067	9.613	0.168
DH02	N24.07.60 E118.24.71	0.592	0.020	9.244	0.285
DH03	N20.22.07 E112.19.64	0.619	0.019	10.532	0.261
NH09	N19.05.92 E110.58.47	0.552	0.038	12.685	0.307
NH13	N18.51.01 E113.18.96	0.673	0.017	9.747	0.257
NH20	N18.30.37 E110.19.33	0.491	0.008	11.203	0.309
NH23	N17.08.93 E112.15.54	0.567	0.019	8.655	0.104
NH31	N17.49.63 E108.31.49	0.574	0.024	11.199	0.308
NH39	N18.26.43 E108.18.35	0.632	0.011	11.304	0.295
NH46	N17.57.27 E109.59.37	0.426	0.008	11.604	0.306
NH50	N17.27.18 E109.25.79	0.681	0.006	13.475	0.351
NH53	N17.32.77 E111.58.88	0.741	0.009	9.357	0.168
NH61	N18.16.17 E111.14.60	0.635	0.021	9.929	0.229
NH66	N17.54.06 E108.49.02	0.618	0.025	9.901	0.215

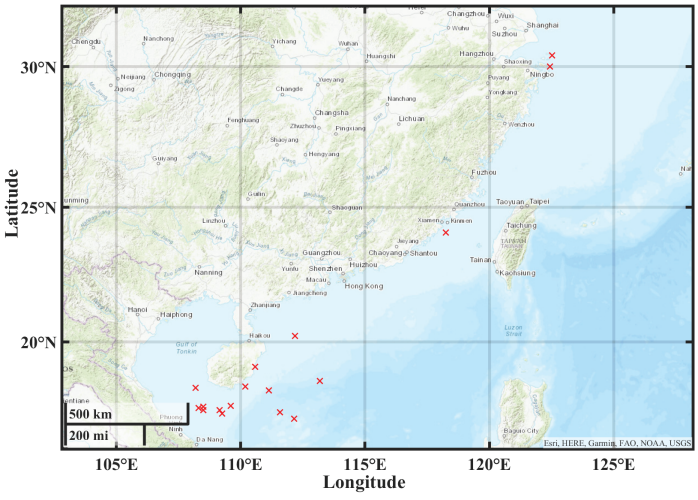


Figure 4. The study area (3 stations in the East China Sea area and 13 stations in the South China Sea area around Hainan Island). In the figure, the red × is represent our observation sites).

4. Results and Discussion

4.1. Data Analysis of Marine Field Observations

In total, 17 simultaneous observations were made by the UVDRAMS in the East China Sea and South China Sea, and 14 effective samples of the UV bispectral RS reflectance were obtained from sampling stations. To ensure the stability of data quality, 10 sets of observation experiments were carried out at each sampling site. In each window time, we will continuously conduct 10 sets of experiments to obtain data, with an interval of 2 min each time, and 10 sets of observation data were obtained and 100 image pixels (pixels 250–350) in the middle of the field of view of the UVDRAMS were selected for analysis to reduce the incident energy inhomogeneity caused by the opening problem at both edges of the field of view. The following analysis was also based on these image points.

Taking the observation data of station NH50 (17°27′11″ N, 109°25′47″ E), a typical station located in the northern part of the South China Sea, as an example, the measured UV spectral radiance associated with the ocean water body is shown in Figure 5.

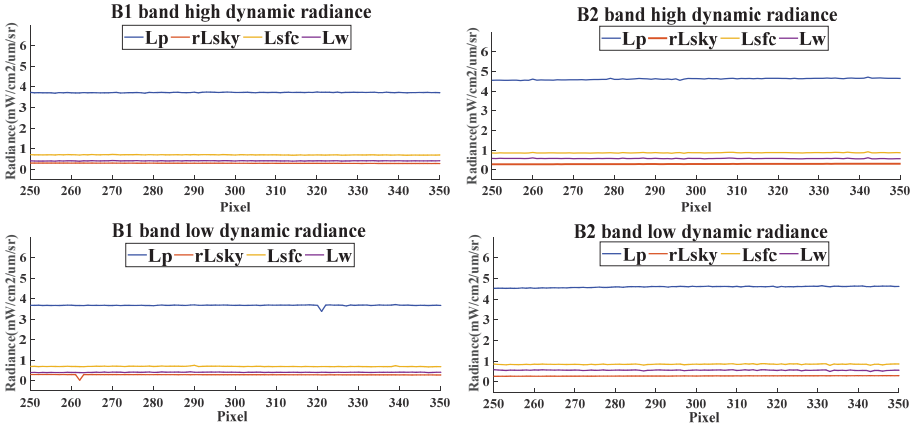


Figure 5. The UV spectral radiance of NH50.

The RS reflectance curves of the NH50 station, calculated from the RS reflectance model analysis, are shown in Figure 6.

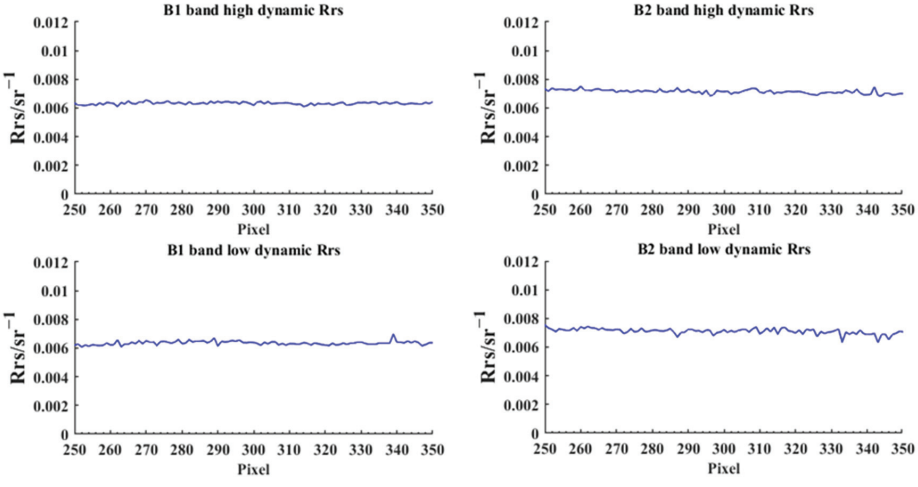


Figure 6. The RS reflectance of NH50.

The test results showed that the average RS reflectance of the NH50 station was 0.063 sr^{-1} in the B1 spectral band and 0.007 sr^{-1} in the B2 spectral band. For the single-observation data, the RS reflectance varied little among the pixels with good consistency.

The water observation results of all the observation stations are demonstrated in Table 2, where the geographic latitude and longitude of each station and the L_{sfc} and L_{sky} required in the transmission model are included.

4.2. Analysis of the Synchronous Observation Data with the UVI Load

Based on the distribution of the sea observation stations shown in Figure 4, the simultaneous observation data of satellite loads at multiple stations were analyzed. The on-satellite observation data of the large clear area with less cloud coverage around the stations were selected according to the precise latitude and longitude information, and 10 groups of DN values from 100 × 100 pixel positions within the sensor's field of view in the relevant sea area were selected, and L_{sat} was calculated. The original in-orbit UV images of sea observation with the HY-1C UVI load are shown in Figure 7.

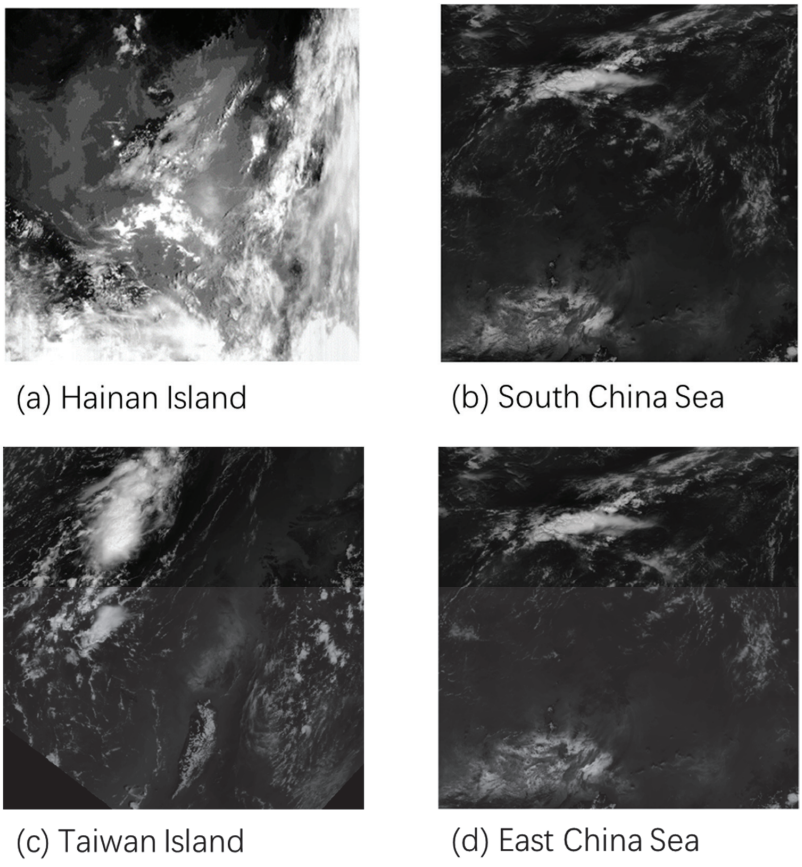


Figure 7. The UV image of the sea surface.

The on-satellite pupil radiance L_{sat} of the sea surface in the cloud-free area around Hainan Island was selected and analyzed. The statistical results of these 100 images show that the average radiance of L_{sat} is 5.964 $mW{\cdot}cm^{-2}{\cdot}um^{-1}{\cdot}sr^{-1}$, and the standard deviation of L_{sat} is 0.983.

The distribution results are shown in Figure 8. It was found that the distribution of pupil radiance in the clear and cloud-free environment obtained from the satellite load observations was highly concentrated.

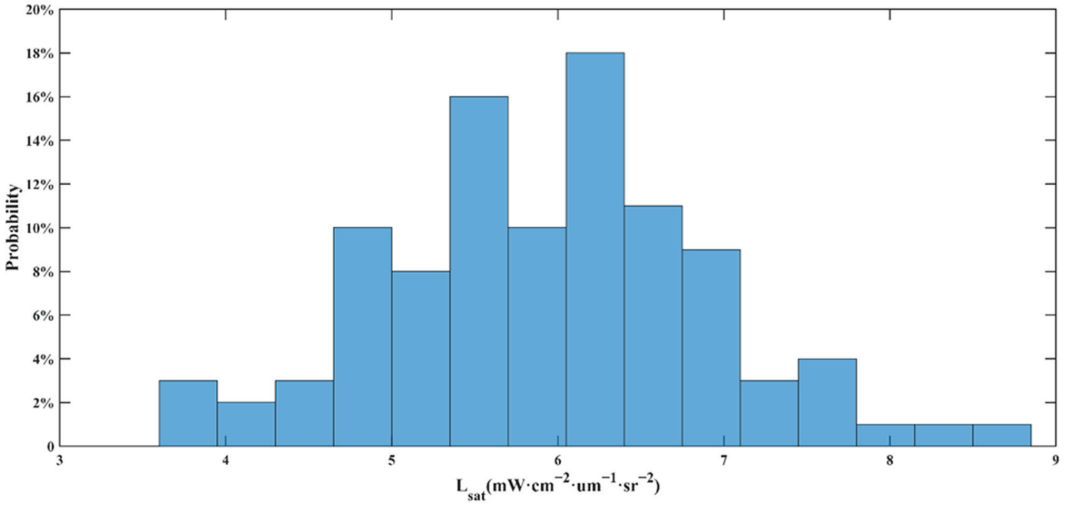


Figure 8. The distribution results of UVI.

4.3. Analysis of the Satellite–Ground Synchrotron Radiation Data Based on the SOA Algorithm

The 140 sets of upward radiance data from all 14 observation stations and the corresponding UVI load radiance data received at the entrance pupil of the synchronous transit sensor are shown in Figure 9.

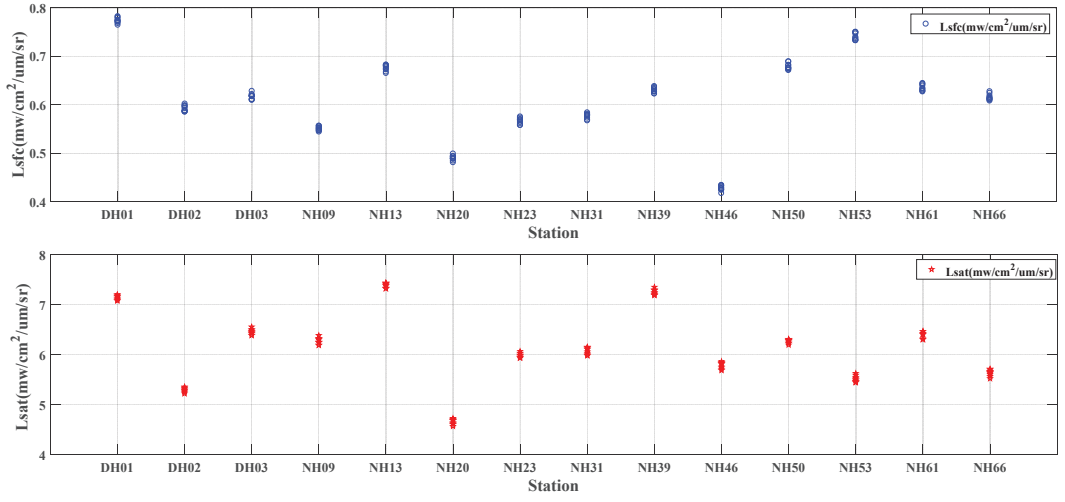


Figure 9. Sea surface radiance data of 14 observation stations and corresponding UVI load pupil radiance data.

In the data analysis of this experiment, the average value of the data of each site was calculated, the surface upward radiance data from 12 of the 14 observation stations and the radiance data received by the pupil of the UVI load synchrotron transit sensor were selected, and the SOA algorithm was employed for iterative fitting.

Assuming that the population size of the SOA algorithm is 50, the maximum number of iterations is 50, the space dimension is 2, the maximum membership degree is 0.95, the minimum membership degree is 0.0111, the maximum weight is 0.9, and the minimum weight is 0.1 (Equation (2)). The range of a and b is [0, 10]. The core problem of the optimization algorithm is to select the objective function:

$$F = \sqrt{\frac{\sum(L_{sat}(i) - L_{sat}{}'(i))^2}{N}} \tag{15}$$

where F is the root mean square error between the model L_{sat} and the actual L_{sat}, and N is the number of data samples. The change curve of the objective function based on the number of iterations is shown in Figure 10.

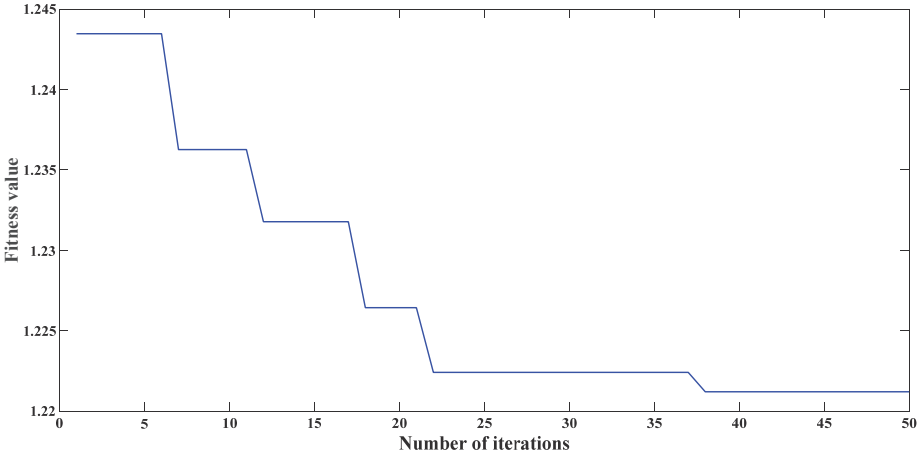


Figure 10. The change curve of the objective function based on the number of iterations.

The best fitting value of F is 1.2212, and thus, the optimal solution is a = 0.6002 and b = 5.7993.

Then, the synchrotron radiation transmission model could be obtained as shown in Equation (16):

$$L_{sat} = 0.6002 \times L_{sfc} + 5.7993 \tag{16}$$

where the atmospheric transmittance is 60.02% and the atmospheric reflected radiance is 5.7993 mW·cm^{-2}·um^{-1}·sr^{-1}.

The distribution and fitted curves of the raw sea surface radiance data observed by the UVDRAMS are shown in Figure 11.

The radiance data from the other two stations, station NH09 and station NH23, were analyzed for validation analysis. The original radiance data, validation radiance data distribution, and fitted curves of UVDRAMS observations are shown in Figure 12.

From Figure 11, we calculated that the coefficient of determination R-squared of the fitted line is 0.4719, the Pearson correlation coefficient is 0.69, and the root mean square error (RMSE) is 0.1456. The Pearson correlation coefficient, also known as the simple correlation coefficient, is used to study the degree of linear correlation between variables and quantitatively describes the degree of correlation between variables. In this paper, we calculated the Pearson correlation coefficient to be 0.69. In statistics, we generally regard the correlation coefficient between 0.6 and 0.8 as a strong correlation, which verifies the validity of our fitting curve.

In addition to the SOA algorithm, we also used several other commonly used heuristic search algorithms, such as genetic algorithm (GA), ant colony optimization (ACO), and

simulated annealing (SA) algorithms, to fit the experimental data and compare the RMSE of the fitted curve, R-squared, and Pearson's r. The results are shown in Table 3. As can be seen from Table 3, the SOA algorithm is the algorithm with the best fitting effect.

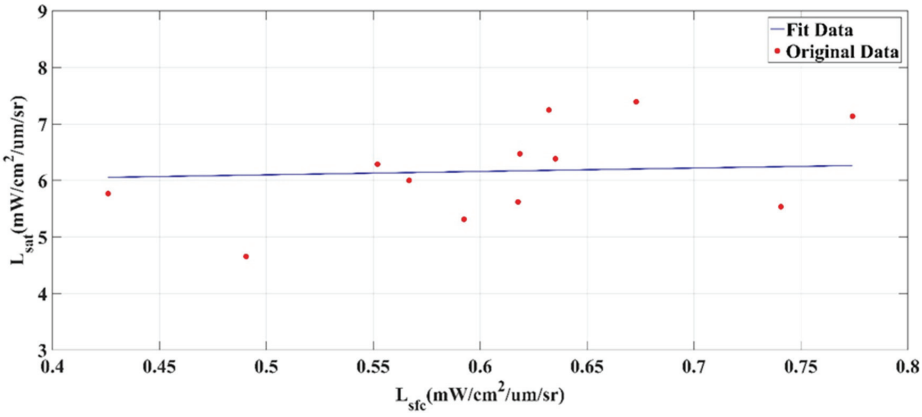


Figure 11. The UVDRAMS data distribution and the fitting curve.

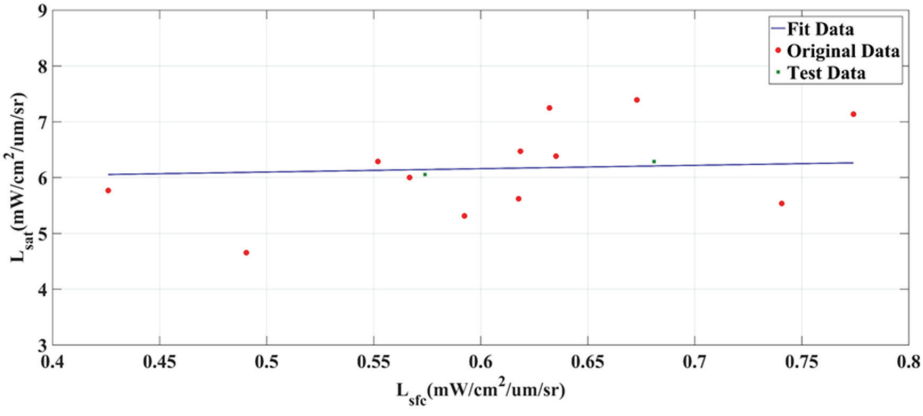


Figure 12. The raw data, validation data distribution, and the fitted curve plots (the coefficient of determination R-squared is 0.4719, Pearson's r is 0.69, and the RMSE is 0.1456).

Table 3. Comparison results of SOA algorithm and other heuristic search algorithms.

	RMSE	R-Squared	Pearson's r
SOA	0.145	0.47	0.69
GA	0.227	0.34	0.58
ACO	0.186	0.38	0.62
SA	0.317	0.28	0.53

In Figure 11, it can be seen that the difference between the measured radiance values of station NH09 and the fitted values of the UV radiative transfer model is 2.7%, and the difference in that for station NH23 is 3.4%, respectively. The results verified the validity of the UV radiative transfer model.

From the fitted straight line in the figure, it can be seen that 97.2% of the incident pupil radiance of the UVI load is obtained due to the contribution of atmospheric reflected radiance, and only 2.8% is obtained from the surface radiation of the water body.

The average standard deviation of the in situ observed radiance at the sea surface is 0.015 $mW \cdot cm^{-2} \cdot um^{-1} \cdot sr^{-1}$, and the inverse variation of the observed data at the water surface is 0.009 $mW \cdot cm^{-2} \cdot um^{-1} \cdot sr^{-1}$ after the attenuation of the atmospheric passage rate. The signal-to-noise ratio of the RS sensor must be at least 640 to effectively distinguish the standard deviation of water body reflectivity in the on-satellite radiation variation, whereas the UVI payload of the HY-1C ocean satellite is designed to have a signal-to-noise ratio of more than 1000 to meet the observation requirements.

Authors should discuss the results and how they can be interpreted from the perspective of previous studies and the working hypotheses. The findings and their implications should be discussed in the broadest context possible. Future research directions may also be highlighted.

5. Conclusions

To improve the quantification of ocean observation technology and support the application of RS data of ocean surface radiance from the HY-1C oceanographic satellite's ultraviolet imager (UVI) payload, a veracity check study of RS data of ocean surface radiance from the UVI payload was conducted.

Using the ocean surface radiance data from 14 stations in the study area that were obtained with the UVDRAMS, as well as the UVI load synchronous observation radiance data combined with the SOA algorithm for identification, optimization, and fitting, a satellite-to-ground synchrotron radiation transfer model was obtained.

The model shows that the coefficient of determination between the fitted curve and the actual observed value is 0.4719, and the square root mean error (RMSE) is 0.1456. The difference between the in situ observed ocean surface radiance values at the two validation sites and the modeled radiance values is 2.7% and 3.4%, respectively, which verifies the validity of the satellite–ground synchrotron radiation transport model.

Our study shows that 97.2% of the incident radiance of the UVI payload is contributed by the atmospheric reflected radiance, and only 2.8% is from the real radiation on the surface of the water body. The signal-to-noise ratio index of >1000 of the HY-1C ocean satellite's UVI payload can effectively distinguish the standard deviation of the reflectivity of the water body in the on-satellite radiation variation, which fully meets the observation requirements. This paper provides preliminary quantified baseline data and a sea surface UV synchrotron radiation measurement solution for verifying the UVI payload of the HY-1C ocean satellite platform and lays a technical foundation for further quantified research in the future.

Author Contributions: Conceptualization, L.L. and Q.L.; methodology, L.L., Q.L. and Q.Z. software, L.L.; validation, L.L.; formal analysis, L.L.; investigation, L.L.; resources, L.L. and D.Y.; data curation, L.L.; writing—original draft preparation, L.L.; writing—review and editing, Q.L., Q.Z., D.Y. and Z.M.; visualization, Q.Z. and L.L.; supervision, D.Y. and Z.M.; project administration, D.Y.; funding acquisition, D.Y. 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 (NSFC) grant number 12103075.

Data Availability Statement: Data available in a publicly accessible repository that does not issue DOIs. Publicly available datasets were analyzed in this study. The HY-1C data can be found here: [https://osdds.nsoas.org.cn/].

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

References

1. Emery, B.; Camps, A. *Introduction to Satellite Remote Sensing: Atmosphere, Ocean, Land and Cryosphere Applications*; Elsevier: Amsterdam, The Netherlands, 2017.
2. Amani, M.; Mehravar, S.; Asiyabi, R.M.; Moghimi, A.; Ghorbanian, A.; Ahmadi, S.A.; Ebrahimy, H.; Moghaddam, S.H.A.; Naboureh, A.; Ranjgar, B. Ocean Remote Sensing Techniques and Applications: A Review (Part II). *Water* **2022**, *14*, 3401. [CrossRef]

3. Amani, M.; Moghimi, A.; Mirmazloumi, S.M.; Ranjgar, B.; Ghorbanian, A.; Ojaghi, S.; Ebrahimy, H.; Naboureh, A.; Nazari, M.E.; Mahdavi, S. Ocean Remote Sensing Techniques and Applications: A Review (Part I). *Water* **2022**, *14*, 3400. [CrossRef]
4. Werdell, P.J.; McKinna, L.I.; Boss, E.; Ackleson, S.G.; Craig, S.E.; Gregg, W.W.; Lee, Z.; Maritorena, S.; Roesler, C.S.; Rousseaux, C.S. An overview of approaches and challenges for retrieving marine inherent optical properties from ocean color remote sensing. *Prog. Oceanogr.* **2018**, *160*, 186–212. [CrossRef]
5. Singh, R.K.; Shanmugam, P. A multidisciplinary remote sensing ocean color sensor: Analysis of user needs and recommendations for future developments. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 5223–5238. [CrossRef]
6. Wu, X.; Xiao, Q.; Wen, J.; You, D.; Hueni, A. Advances in quantitative remote sensing product validation: Overview and current status. *Earth-Sci. Rev.* **2019**, *196*, 102875. [CrossRef]
7. Jin, M.; Ji, Z.; Shaomin, L.; Yujia, W. Review on validation of remotely sensed land surface temperature. *Adv. Earth Sci.* **2017**, *32*, 615.
8. Jiang, H. Indirect validation of ocean remote sensing data via numerical model: An example of wave heights from altimeter. *Remote Sens.* **2020**, *12*, 2627. [CrossRef]
9. Arun Kumar, S.; Babu, K.; Shukla, A. Comparative analysis of chlorophyll-a distribution from SEAWIFS, MODIS-AQUA, MODIS-TERRA and MERIS in the Arabian Sea. *Mar. Geod.* **2015**, *38*, 40–57. [CrossRef]
10. Bartlett, B.; Casey, J.; Padula, F.; Pearlman, A.; Pogorzala, D.; Cao, C. Independent validation of the advanced baseline imager (ABI) on NOAA's GOES-16: Post-launch ABI airborne science field campaign results. In Proceedings of the Earth Observing Systems XXIII, San Diego, CA, USA, 19–23 August 2018; pp. 143–157.
11. McCorkel, J.; Efremova, B.; Hair, J.; Andrade, M.; Holben, B. GOES-16 ABI solar reflective channel validation for earth science application. *Remote Sens. Environ.* **2020**, *237*, 111438. [CrossRef]
12. Yang, J.; Zhang, J. Validation of Sentinel-3A/3B satellite altimetry wave heights with buoy and Jason-3 data. *Sensors* **2019**, *19*, 2914. [CrossRef]
13. Urabe, T.; Okamura, Y.; Tanaka, K.; Mokuno, M. In-orbit commissioning activities results of GCOM-C/SGLI. In Proceedings of the Sensors, Systems, and Next-Generation Satellites XXII, Berlin, Germany, 10–13 September 2018; pp. 88–107.
14. Nakajima, T.Y.; Ishida, H.; Nagao, T.M.; Hori, M.; Letu, H.; Higuchi, R.; Tamaru, N.; Imoto, N.; Yamazaki, A. Theoretical basis of the algorithms and early phase results of the GCOM-C (Shikisai) SGLI cloud products. *Prog. Earth Planet. Sci.* **2019**, *6*, 52. [CrossRef]
15. Tanaka, K.; Okamura, Y.; Mokuno, M.; Amano, T.; Yoshida, J. First year on-orbit calibration activities of SGLI on GCOM-C satellite. In Proceedings of the Earth Observing Missions and Sensors: Development, Implementation, and Characterization V, Honolulu, HI, USA, 24–26 September 2018; pp. 101–110.
16. Urabe, T.; Ando, S.; Okamura, Y.; Tanaka, K.; Mokuno, M.; Amano, T.; Shiratama, K.; Yoshida, J. Pre-launch instrument characterization results and in-orbit verification plan of GCOM-C/SGLI. In Proceedings of the Sensors, Systems, and Next-Generation Satellites XXI, Warsaw, Poland, 11–14 September 2017; pp. 130–143.
17. Tian, H.; Huang, N.; Niu, Z.; Qin, Y.; Pei, J.; Wang, J. Mapping winter crops in China with multi-source satellite imagery and phenology-based algorithm. *Remote Sens.* **2019**, *11*, 820. [CrossRef]
18. Tian, H.; Pei, J.; Huang, J.; Li, X.; Wang, J.; Zhou, B.; Qin, Y.; Wang, L. Garlic and winter wheat identification based on active and passive satellite imagery and the google earth engine in northern china. *Remote Sens.* **2020**, *12*, 3539. [CrossRef]
19. Modabberi, A.; Noori, R.; Madani, K.; Ehsani, A.H.; Mehr, A.D.; Hooshyaripor, F.; Kløve, B. Caspian Sea is eutrophying: The alarming message of satellite data. *Environ. Res. Lett.* **2020**, *15*, 124047. [CrossRef]
20. Shanshan, M. A LM-2C Launches HY-1C Satellite. *Aerosp. China* **2018**, *19*, 59.
21. Suo, Z.; Lu, Y.; Liu, J.; Ding, J.; Xing, Q.; Yin, D.; Xu, F.; Liu, J. HY-1C ultraviolet imager captures algae blooms floating on water surface. *Harmful Algae* **2022**, *114*, 102218. [CrossRef] [PubMed]
22. Suo, Z.; Lu, Y.; Liu, J.; Ding, J.; Yin, D.; Xu, F.; Jiao, J. Ultraviolet remote sensing of marine oil spills: A new approach of Haiyang-1C satellite. *Opt. Express* **2021**, *29*, 13486–13495. [CrossRef] [PubMed]
23. Dai, C.; Chen, W.; Song, Y.; Zhu, Y. Seeker optimization algorithm: A novel stochastic search algorithm for global numerical optimization. *J. Syst. Eng. Electron.* **2010**, *21*, 300–311. [CrossRef]
24. Zhang, R.; Tian, J.; Li, Z.; Su, H.; Chen, S.; Tang, X. Principles and methods for the validation of quantitative remote sensing products. *Sci. China Earth Sci.* **2010**, *53*, 741–751. [CrossRef]
25. Olteţeanu, A.-M.; Zunjani, F.H. A visual remote associates test and its validation. *Front. Psychol.* **2020**, *11*, 26. [CrossRef]
26. Van Etten, A.; Lindenbaum, D.; Bacastow, T.M. Spacenet: A remote sensing dataset and challenge series. *arXiv* **2018**, arXiv:1807.01232.
27. Xue, J.; Su, B. Significant remote sensing vegetation indices: A review of developments and applications. *J. Sens.* **2017**, *2017*, 1353691. [CrossRef]
28. Ruiz-Arias, J.A. Spectral integration of clear-sky atmospheric transmittance: Review and worldwide performance. *Renew. Sustain. Energy Rev.* **2022**, *161*, 112302. [CrossRef]
29. Kobayashi, S.; Sanga-Ngoie, K. A comparative study of radiometric correction methods for optical remote sensing imagery: The IRC vs. other image-based C-correction methods. *Int. J. Remote Sens.* **2009**, *30*, 285–314. [CrossRef]
30. Zibordi, G.; Voss, K.; Johnson, B.; Mueller, J. *Protocols for Satellite Ocean Color Data Validation: In Situ Optical Radiometry*; IOCCG Protocols Series; IOCCG: Dratmouth, NS, Canada, 2019.

31. Duan, S.; Luo, H.; Liu, H. A Multi-Strategy Seeker Optimization Algorithm for Optimization Constrained Engineering Problems. *IEEE Access* **2022**, *10*, 7165–7195. [CrossRef]
32. Shafik, M.B.; Chen, H.; Rashed, G.I.; El-Sehiemy, R.A. Adaptive multi objective parallel seeker optimization algorithm for incorporating TCSC devices into optimal power flow framework. *IEEE Access* **2019**, *7*, 36934–36947. [CrossRef]
33. Choudhury, S.; Dash, T.P.; Bhowmik, P.; Rout, P.K. A novel control approach based on hybrid Fuzzy Logic and Seeker Optimization for optimal energy management between micro-sources and supercapacitor in an islanded Microgrid. *J. King Saud Univ.-Eng. Sci.* **2020**, *32*, 27–41. [CrossRef]
34. Cui, H.; Guan, Y.; Chen, H.; Deng, W. A novel advancing signal processing method based on coupled multi-stable stochastic resonance for fault detection. *Appl. Sci.* **2021**, *11*, 5385. [CrossRef]
35. Kumar, D.; Samantaray, S. Implementation of multi-objective seeker-optimization-algorithm for optimal planning of primary distribution systems including DSTATCOM. *Int. J. Electr. Power Energy Syst.* **2016**, *77*, 439–449. [CrossRef]
36. Ruddick, K.G.; Voss, K.; Boss, E.; Castagna, A.; Frouin, R.; Gilerson, A.; Hieronymi, M.; Johnson, B.C.; Kuusk, J.; Lee, Z. A review of protocols for fiducial reference measurements of water-leaving radiance for validation of satellite remote-sensing data over water. *Remote Sens.* **2019**, *11*, 2198. [CrossRef]
37. Mueller, R.; Trentmann, J.; Träger-Chatterjee, C.; Posselt, R.; Stöckli, R. The role of the effective cloud albedo for climate monitoring and analysis. *Remote Sens.* **2011**, *3*, 2305–2320. [CrossRef]
38. Liu, Y.; Wu, W.; Jensen, M.P.; Toto, T. Relationship between cloud radiative forcing, cloud fraction and cloud albedo, and new surface-based approach for determining cloud albedo. *Atmos. Chem. Phys.* **2011**, *11*, 7155–7170. [CrossRef]
39. Harrison, E.F.; Minnis, P.; Barkstrom, B.; Ramanathan, V.; Cess, R.; Gibson, G. Seasonal variation of cloud radiative forcing derived from the Earth Radiation Budget Experiment. *J. Geophys. Res. Atmos.* **1990**, *95*, 18687–18703. [CrossRef]
40. Ramanathan, V.; Cess, R.; Harrison, E.; Minnis, P.; Barkstrom, B.; Ahmad, E.; Hartmann, D. Cloud-radiative forcing and climate: Results from the Earth Radiation Budget Experiment. *Science* **1989**, *243*, 57–63. [CrossRef] [PubMed]

electronics

MDPI

Article

CNTR-YOLO: Improved YOLOv5 Based on ConvNext and Transformer for Aircraft Detection in Remote Sensing Images

Fengyun Zhou, Honggui Deng *, Qiguo Xu and Xin Lan

School of Physics and Electronics, Central South University, Lushan South Road, Changsha 410083, China; 212212102@csu.edu.cn (F.Z.); 202211045@csu.edu.cn (Q.X.); 212212063@csu.edu.cn (X.L.)
* Correspondence: denghonggui@csu.edu.cn; Tel.: +86-199-7499-4797

Abstract: Aircraft detection in remote sensing images is an important branch of target detection due to the military value of aircraft. However, the diverse categories of aircraft and the intricate background of remote sensing images often lead to insufficient detection accuracy. Here, we present the CNTR-YOLO algorithm based on YOLOv5 as a solution to this issue. The CNTR-YOLO algorithm improves detection accuracy through three primary strategies. (1) We deploy DenseNet in the backbone to address the vanishing gradient problem during training and enhance the extraction of fundamental information. (2) The CBAM attention mechanism is integrated into the neck to minimize background noise interference. (3) The C3CNTR module is designed based on ConvNext and Transformer to clarify the target's position in the feature map from both local and global perspectives. This module is applied before the prediction head to optimize the accuracy of prediction results. Our proposed algorithm is validated on the MAR20 and DOTA datasets. The results on the MAR20 dataset show that the mean average precision (mAP) of CNTR-YOLO reached 70.1%, which is a 3.3% improvement compared with YOLOv5l. On the DOTA dataset, the results indicate that the mAP of CNTR-YOLO reached 63.7%, which is 2.5% higher than YOLOv5l.

Keywords: remote sensing images; aircraft detection; YOLOv5; ConvNext; Transformer

Citation: Zhou, F. ; Deng, H.; Xu, Q.; Lan, X. CNTR-YOLO: Improved YOLOv5 Based on ConvNext and Transformer for Aircraft Detection in Remote Sensing Images. *Electronics* **2023**, *12*, 2671. https://doi.org/10.3390/electronics12122671

Academic Editor: Gerardo Di Martino

Received: 15 May 2023
Revised: 12 June 2023
Accepted: 12 June 2023
Published: 14 June 2023

1. Introduction

With the help of advanced satellite remote sensing technology, many high-resolution remote sensing images have been produced, which often contain a wealth of information. These images also provide rich materials for the research of target detection, so the detection methods of remote sensing targets have become a hot topic for scholars [1,2]. Among all types of targets, aircraft have high mobility and are of great value in various fields, especially in the military. Therefore, studying the detection methods of aircraft targets in remote sensing images is significant. However, it is still a challenging task because of the top–down view of remote sensing images, which can only acquire the upper surface features of objects, and due to many aircraft types being highly similar to each other, as well as satellite photography being susceptible to external factors such as weather, light, shadows and so on [3,4].

In recent years, deep learning algorithms have become the prevailing method for target detection due to advances in computer techniques. Target detection using deep learning algorithms can be categorized into two types: single-stage target detection algorithm, and two-stage target detection algorithm. The single-stage algorithm treats target detection as a combination of regression and classification tasks, while the two-stage algorithm first generates a collection of candidate regions and then identifies and classifies the target object based on these regions [5]. Two-stage algorithms, including R-CNN [6], Fast R-CNN [7], Faster R-CNN [8], and Cascade R-CNN [9], tend to have higher accuracy but suffer from high computational requirements due to the large number of candidate frames, leading to lengthy training periods and slow detection speeds. In contrast, the detection accuracy

of single-stage algorithms is typically lower than the two-stage algorithms; however, the detection speed is substantially faster. Notable examples of single-stage algorithms are YOLO [10], SSD [11], Retinanet [12], and FCOS [13].

Numerous studies have explored the application of deep learning algorithms to detect aircraft targets in remote sensing images. For instance, Liu proposed a two-stage algorithm that utilizes the Harris operator to detect corners, clusters them using mean drift clustering to generate small yet precise candidate regions, and subsequently identifies the aircraft's region by leveraging a CNN model, resulting in enhanced detection accuracy [14]. In the DPANet, Shi introduced a deconvolution module to extract external structural features of the aircraft, which was followed by a position attention mechanism to extract internal structural features, which reduced the false detection rate and improved detection precision [15]. Wu optimized Mask R-CNN by combining self-calibrated convolution with ResNet in the backbone, thus making the features more discriminative and resulting in improved network accuracy [16]. For his part, Ji expanded on Fast R-CNN by incorporating a multi-angle change module that extracts target features from multiple viewpoints, thereby reducing the false detection rate. Furthermore, he employed a box detection post-processing method with a majority voting strategy to further minimize the likelihood of misjudgment [17]. Although these algorithms are two-stage and possess unique accuracy advantages, they are still more complex relative to one-stage algorithms. Therefore, many researchers continue to focus on one-stage algorithms, particularly based on the YOLO series. For example, Cao improved the YOLOv3 model by adding a detection scale with a smaller perceptual field and using L2 regularization to combat overfitting [18]. Zhou devised the Deeper and Wider Module (DAWM), which drew inspiration from the Inception–ResNet model. Incorporating the DAWM architecture into YOLOv3 effectively mitigated the impact of background noise and further advanced network performance [19]. Luo added center and scale calibration at the beginning and end of the batch normalization layer in YOLOv5 to address the problem that the batch normalization layer ignores the representation differences between instances, enabling features to be corrected, which has improved the performance of the overall network [20]. Liu proposed the YOLO-extract algorithm, which removed feature layers and prediction heads in YOLOv5 with suboptimal feature extraction ability and replaced them with a new feature extractor possessing stronger feature extraction capabilities. This modification resulted in improved accuracy and reduced computational costs [21]. Notwithstanding the above advances in aircraft target detection algorithms, some algorithms fail to fully utilize global and local information of remote sensing images, resulting in aircraft target misdetection. To address this shortcoming, we require a novel aircraft target detection algorithm for remote sensing images that leverages global and local information more efficiently.

In this paper, we present CNTR-YOLO, which is an improved version of YOLOv5. We have made several modifications to enhance network performance. Firstly, we introduced the Dense module based on DenseNet to reinforce the feature extraction capability of the backbone. By reusing features, this module mitigates the loss of valid information. Secondly, we added the CBAM attention module to the neck to produce attention maps iteratively across both channel and spatial dimensions. This module assists in identifying areas with aircraft targets in images while reducing the impact of background noise interference. Lastly, in order to make full use of global and local information in remote sensing images, we established the C3CNTR module by combining the Transformer Block and ConvNext Block. This novel design is placed before the detection head of YOLOv5 and leverages the Transformer Block for processing global information and the ConvNext Block for processing local information.

Our contributions can be summarized as follows:

1. We propose a single-stage object detection algorithm to improve the accuracy of aircraft detection in remote sensing images.

2. For the first time, we design a structure that combines a convolutional network and Transformer in YOLOv5 to assist the prediction head, maximizing the utilization of local and global feature information.

3. We validate some effective measures to improve the performance in YOLOv5, such as using DenseNet to improve feature extraction and the CBAM attention mechanism to reduce interference from background information.

2. Related Work

In this section, we provide an overview of the key components of our proposed algorithm. Specifically, we discuss YOLOv5, Transformer, and ConvNext.

2.1. YOLOv5

YOLOv5 was released in 2020 by Ultralytics LLC and was built upon the foundation of YOLOv3 [22]. YOLOv5 rectified the earlier issue of faster detection speed at the expense of accuracy. It also improved real-time performance and simplified the network structure. Comprised of a backbone, neck, and head, YOLOv5 features five models, ranging from YOLOv5n to YOLOv5x based on the network depth. Despite YOLOv5x exhibiting marginally superior detection accuracy compared to YOLOv5l, the latter delivers faster speeds and requires fewer hardware resources. Therefore, we conduct research based on YOLOv5l.

Figure 1 illustrates the architecture of YOLOv5. The feature extraction network of YOLOv5 is composed of a CSPDarkNet53 network [23] and an SPPF layer. The neck utilizes a PANet [24] structure, and the head is a YOLO detection head that comprises a convolution layer and a prediction component. In YOLOv5, the C3 module is one of the most frequently applied modules. The structure of the C3 module, as shown in Figure 2, consists of three convolutional modules and a Bottleneck. The Bottleneck is a residual block that possesses faster computation speeds than the residual block of ResNet [25]. Furthermore, it enables a deeper network architecture while reducing computational parameters.

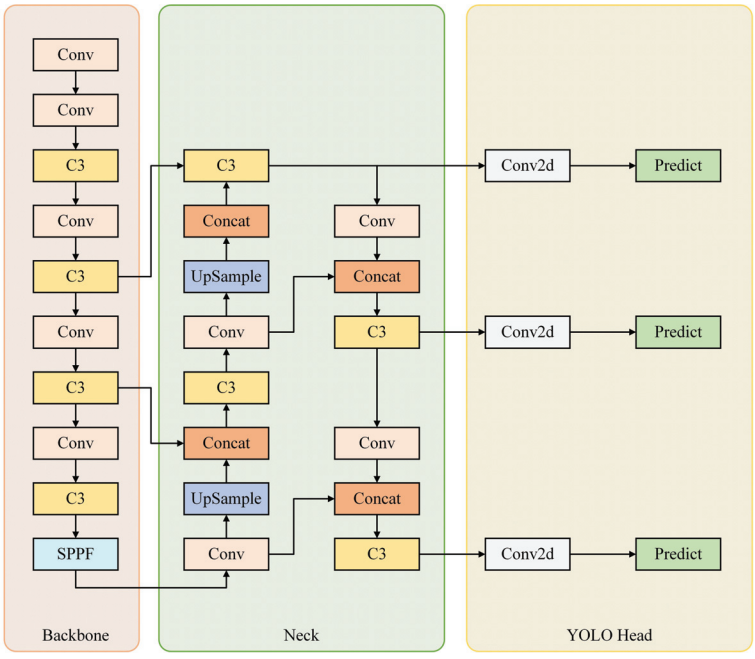


Figure 1. The architecture of YOLOv5.

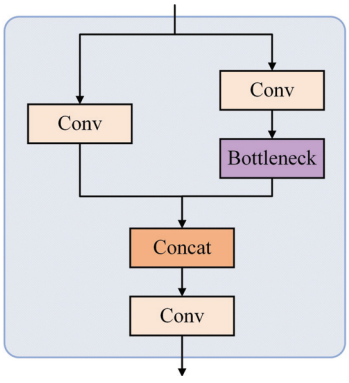


Figure 2. The structure of C3 module.

While YOLOv5 has demonstrated excellent performance across various vision tasks, its direct application to aircraft target detection in remote sensing images falls short of satisfactory outcomes. Thus, this paper introduces several improvements to enhance its performance in this domain.

2.2. *Transformer*

In recent years, Transformer [26] has achieved significant success in the field of natural language processing (NLP). As the size of the convolutional kernel constrains its ability to acquire local representations, researchers have looked to extend Transformer's functionality to computer vision. To this end, Dosovitskiy et al. proposed the Vision Transformer (ViT) methodology [27], which leverages Multiple Self-Attention (MSA) to capture long-range feature dependencies within internal information.

The details of the ViT methodology can be succinctly summarized as follows. Firstly, a two-dimensional image is converted into several one-dimensional sequences. Location encoding is then incorporated to provide information on the image's spatial position. Subsequently, the sequences, with learnable location encoding, are passed through the Transformer encoder, which calculates global attention and extracts features via the multi-headed attention module. Lastly, the MLP layer yields the prediction categories.

Several researchers have already integrated Transformer with YOLOv5. For example, in the detection of targets during UAV shooting scenes, Zhu replaced the Bottleneck in the C3 structure of the original YOLOv5 with the Transformer Block to create the C3TR module [28]. Figure 3 displays the structure of the C3TR module. Transformer's unique properties enable the C3TR module to capture global information and abundant contextual information from features.

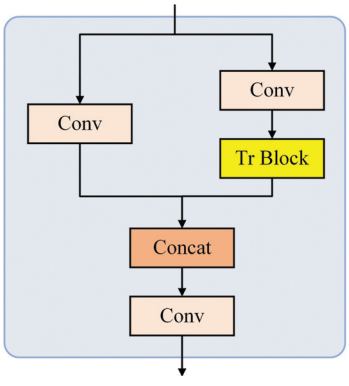


Figure 3. The structure of the C3TR module; Tr Block stands for Transformer Block.

Target detection in remote sensing images presents unique challenges compared to UAV shooting scenes, including larger shooting distances, smaller objects, and a single angle of aircraft targets (which are mostly vertical). Given these difficulties, it is crucial to explore alternative approaches to integrate Transformer and address these complexities.

2.3. ConvNext

In the realm of computer vision, ViT has swiftly replaced convolutional networks as the state-of-the-art approach for image classification models. On the other hand, FAIR's ConvNext [29], which relies entirely on standard convolutional networks, offers comparable accuracy and generalizability to Transformer.

ConvNext does not introduce significant innovations to the overall network architecture or construction ideas. Instead, it makes some modifications to the existing ResNet network by incorporating some advanced concepts of Transformer. These changes aim to combine the advantages of both convolutional neural networks (CNNs) and Transformer networks, which ultimately leads to improved CNN performance.

In contrast to Transformer, ConvNext, built using convolutional networks, exhibits a greater capacity to capture local information. This ability plays a pivotal role in detecting high-resolution remote sensing images. The present study proposes a novel joint design that integrates the strengths of both Transformers and ConvNext to improve detection performance.

3. Theoretical Model

To address the challenges associated with detecting aircraft targets in remote sensing images, we developed CNTR-YOLO based on YOLOv5. In this section, we first present the architecture of CNTR-YOLO. Subsequently, we elaborate on the critical components of CNTR-YOLO, including the C3CNTR module, Dense module, and CBAM attention module.

3.1. Overview of CNTR-YOLO

The architecture of the proposed CNTR-YOLO module is shown in Figure 4. Compared with YOLOv5, CNTR-YOLO has a total of seven differences. First, we replaced a C3 module with a Dense module at the end of the Backbone, then inserted a CBAM attention module after each of the first three C3 modules in the neck, and finally, the C3CNTR module is inserted before the detection head.

3.2. C3-ConvNext-Transformer (C3CNTR) Module

To enhance YOLOv5's understanding of global and local information, we have drawn inspiration from the success of incorporating Transformer in YOLOv5 and designing the C3TR module in reference [28]. In light of this experience, we introduce ConvNext and Transformer to develop the C3CNTR module. ConvNext enhances the utilization of local information, while Transformer improves the utilization of global information. Figure 5 illustrates the structure of the C3CNTR module.

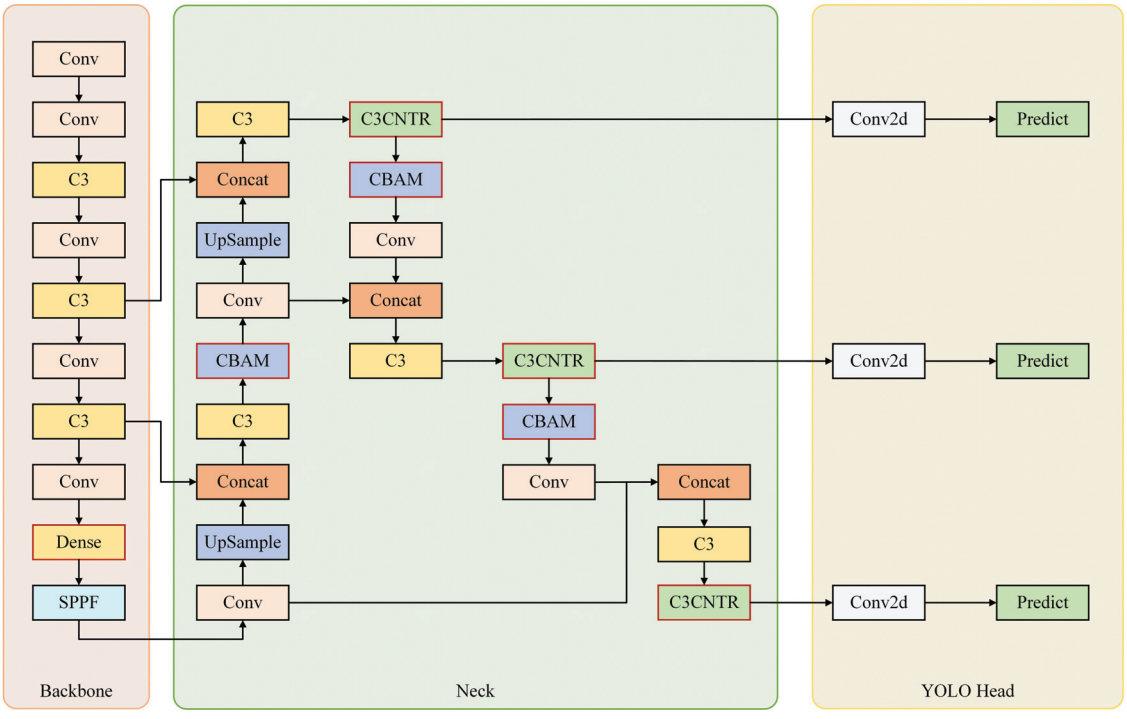


Figure 4. The architecture of CNTR-YOLO.

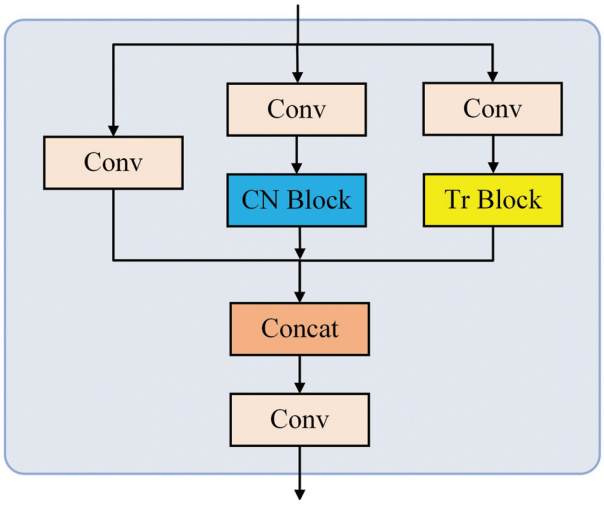


Figure 5. The structure of C3CNTR; CN Block stands for ConvNext Block.

3.2.1. Transformer Block

In the Transformer Block shown in Figure 6, we employ a classic Transformer Encoder architecture. In contrast to the standard convolutional network, this architecture utilizes certain special operations that will be elaborated on shortly.

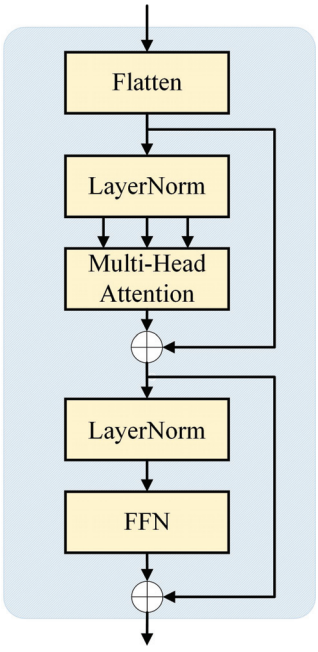


Figure 6. The structure of Transformer Block.

1. Flatten

A Flatten operation is located at the outset of the Transformer Encoder and serves to convert two-dimensional feature maps into one-dimensional sequences of feature maps. Given an input feature map $X \in R^{H \times W \times C}$, it becomes $X' \in R^{N \times C}$ after Flatten, where $N = H \times W$.

2. Multi-head attention

Multi-head attention is a global operation that allows the Transformer Encoder to discover correlation information on a feature's entire range. The feature map undergoes conversion into $Q, K, V \in R^{N \times C}$ with different linear mappings following Flatten and LayerNorm to serve as input for multi-head attention. Comprising several single-head attentions, multi-head attention executes one operation on Q, K, V with each single-head attention. The output expression of the i-th single-head attention is as follows:

$$Output_i = S_i V_i \tag{1}$$

$$S_i = softmax(Q_i K_i^T) \tag{2}$$

where Q_i , K_i , V_i is the multiplication of Q, K, V and the i-th single-head attention's weight matrix, while $S_i \in R^{N \times N}$ represents the attention matrix, revealing the correlation between each element of the feature map and other elements. $Output_i$ refers to the feature that consolidates global information. After each single-head attention completes its operation, the resulting outputs are unified via the concatenation layer. The ultimate output expression is shown as follows:

$$Output_{all} = Concat(Output_1, \ldots, Output_h) \tag{3}$$

where h is the number of multi-head attention heads.

3. FFN

The output of multi-head attention advances to FFN once it undergoes LayerNorm. FFN refers to a Feed-Forward Network that essentially comprises two fully connected layers; one of which has Relu activation, while there is a Dropout between the two layers. The expression for FFN processing is shown below:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{4}$$

where x is the sequence of input feature maps, W_1 and b_1 are the weights and offsets of the first fully connected layer, and W_2 and b_2 are the weights and offsets of the second fully connected layer.

3.2.2. ConvNext Block

The ConvNext Block's structure is shown in Figure 7, which adopts the standard ConvNext network structure.

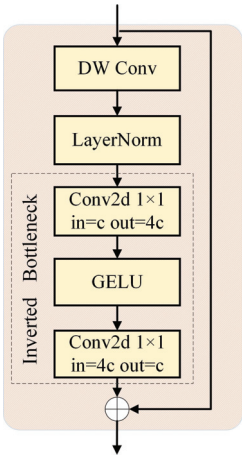


Figure 7. The structure of ConvNext Block.

While ConvNext is essentially a convolutional network, its design delineates some similarities to Transformer, which are elaborated upon below.

1. DW Conv

A group convolution employs multiple groups of convolutional filters for convolution. On the other hand, DWConv (depthwise convolution) refers to a special group convolution in which the number of groups equals the number of channels. Similar to multi-head attention in Transformer, depthwise convolution plays a pivotal role in ConvNext's architecture. Depthwise convolution, akin to the weighted sum operation in multi-head attention, performs operations on a channel-by-channel basis, amalgamating information only in the spatial dimension. The combination of depthwise convolution and 1×1 convolution allows for a separation of the spatial and channel dimensions of the feature maps. Each operation, by mixing information either across the spatial dimension or channel dimension, is performed independently, which is analogous to Transformers. Comprised of only pure convolutional networks, ConvNext's global perceptual field differs from that of Transformers. To compensate for this limitation, ConvNext uses 7×7 convolution kernels in depthwise convolution.

2. Inverted Bottleneck

The ConvNext Block culminates in an inverted bottleneck, which is a design element also found in Transformer. In Transformer Encoder, a crucial design specification entails incorporating an inverted bottleneck at the end, amplifying the hidden dimensions of

the two fully connected layers in FFN to four times the input dimensions. Following the advent of Transformer, various cutting-edge convolutional networks adopted the inverted bottleneck design, such as MobileNetV2 [30]. Similar in approach to Transformer, ConvNext creates the Inverted Bottleneck at the end via two 1×1 convolutions. The role of 1×1 convolution is commensurate to that of a fully connected layer. The first 1×1 convolution expands the input channel four times, while the latter restores the number of input channels. The authors of ConvNext have also validated that this design enhances network performance across multiple tasks, encompassing classification and detection.

3.3. Dense Module

Toward the end of the feature extraction network, we exchanged a C3 module for a Dense module, aiming to heighten the network's efficiency in utilizing feature information. The Dense module follows the structure of C3, as depicted in Figure 8, and it contains the architecture of DenseNet [31], which is delineated in Figure 9.

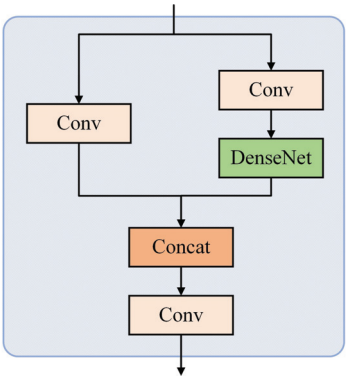


Figure 8. The structure of the Dense module.

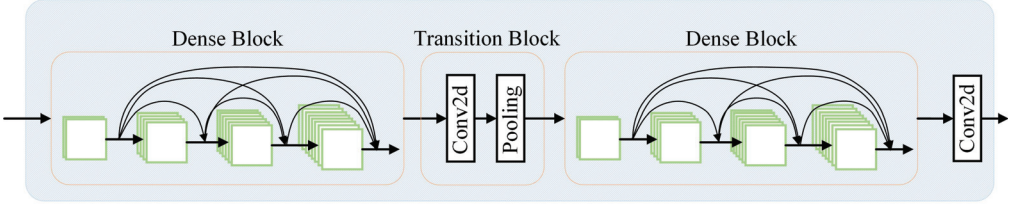


Figure 9. The structure of DenseNet.

DenseNet melds concepts from ResNet and Inception networks [32], possessing four fundamental benefits, comprising: retaining low-latitude features; enhancing feature reuse; mitigating the gradient disappearance problem; and considerably diminishing the number of parameters. Its architecture principally incorporates numerous DenseNet Blocks and Transition Blocks, and we select two and one, respectively, for each. A DenseNet Block with N layers of convolution possesses $N(N+1)/2$ connections, with each layer's input deriving from all previous layers' output, which is a stark contrast to the N connections in a traditional convolutional neural network with N layers. This unique connection methodology in a DenseNet Block optimizes a better utilization of features and obviates the need for learning a considerable mass of irrelevant feature information, thereby preventing gradient explosion and diminishing the likelihood of overfitting. Elevated feature extraction in the network is achieved while reducing computation and the number of parameters.

Assuming N convolution layers exist in a Dense Block, the expression for the n-th layer of output is as follows:

$$x_n = f_n([x_1, x_2, \ldots, x_{n-1}]) \tag{5}$$

where f_n represents the nonlinear operation at the n-th layer, and $[x_1, x_2, \ldots, x_{n-1}]$ represents the operation of concatenating all the outputs before the n-th layer. Concatenation is distinguishable from residual connection, the latter which simply adds the values of two features together. Whereas concatenation, by comparison, increases the number of channels to enable preservation of the previous feature information in its entirety. To ensure consistency in the number of channels of input features across each DenseNet Block, a Transition Block is implemented to restore the number of channels in the output feature from the previous DenseNet Block.

3.4. CBAM

The Convolutional Block Attention Module (CBAM) [33] comprises two sub-modules: the Channel Attention Module (CAM) and the Spatial Attention Module (SAM). Through its attention mechanism, CBAM simultaneously regulates the channel and space features, thus enabling the network to capture a comprehensive range of information contained in the feature map. Illustratively, Figure 10 below depicts the diagram of CBAM.

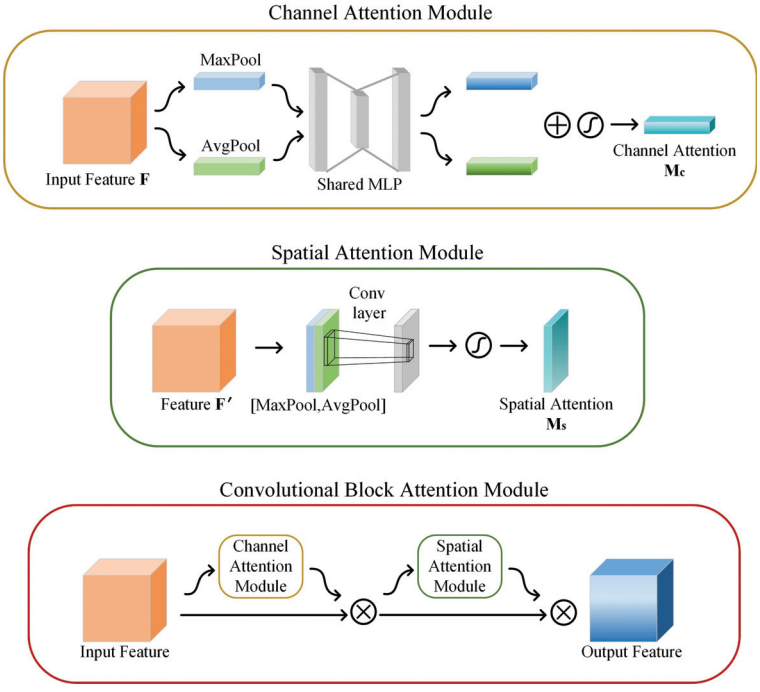


Figure 10. The structure of CBAM attention module.

The input feature map will first pass through the CAM. At the beginning of CAM is a global max pooling layer and a global average pooling layer. These two pooling layers will pool the feature maps based on height and width to obtain two $1 \times 1 \times C$ feature maps (C is the number of channels), and then, the obtained feature maps will be fed into a two-layer MLP network, which is shared by the two input features. The MLP-processed feature maps are summed element-wise, and finally, the sigmoid activation function is used to generate the channel attention feature. The channel attention feature will be multiplied element-wise with the input feature map to obtain the input feature map of SAM.

In SAM, first, the input feature map from CAM will undergo a channel-based global maximum pooling and global average pooling to obtain two $H \times W \times 1$ feature maps; H and W are the height and width of the feature maps, respectively. Then, the two feature maps are concatenated in the channel dimension, and the number of channels of the feature map is doubled. Next, the number of channels of the feature map is reduced by a convolutional layer followed by a sigmoid activation function, which generates a spatial attention feature. Finally, the spatial attention feature is multiplied based on element-wise with the input features of SAM to obtain the final features generated by CBAM.

4. Experiments

In this section, we first introduce the dataset used in the experiments, namely the MAR20 dataset. Subsequently, we also explain the evaluation metrics and implementation details of the experiments. The experiments can be broadly summarized as the comparison of CNTR-YOLO with other algorithms alongside the ablation study.

4.1. Dataset

The MAR20 dataset [34], presently the largest dataset for remote sensing military aircraft target recognition, is utilized in this paper to validate the proposed algorithm's performance. The dataset contains 3842 images and 22,341 instances of mostly 800×800 pixels gathered from 60 military airports across the United States, Russia, and other countries using Google Earth. The MAR20 dataset specifically includes 20 aircraft models, with six of them being the Russian SU-35 fighter, TU-160 bomber, TU-22 bomber, TU-95 bomber, SU-34 fighter-bomber, and SU-24 fighter-bomber. The remaining 14 aircraft models include the U.S. C-130 transport, C-17 transport, C-5 transport, F16 fighter, E-3 AWACS, B-52 bomber, P-3C ASW, B-1B bomber, E-8 joint battlefield surveillance aircraft, F-15 fighter, KC-135 air refueling aircraft, F-22 fighter, F/A-18 combat attack aircraft, and KC-10 air refueling aircraft. These aircraft model types are denoted with abbreviations A1 to A20. The dataset is split into a training set of 1331 images and 7870 instances and a testing set of 2511 images and 14471 instances, as shown in this paper's experimentation.

The DOTA dataset [35] is a large remote sensing image dataset consisting of 2806 high-resolution images obtained from Google Earth and multiple satellite sensors with image sizes ranging from 800×800 pixels to 4000×4000 pixels. In comparison to the MAR20 dataset, DOTA includes a more comprehensive range of object categories, including Plane, Baseball diamond, Bridge, Ground field track, Small vehicle, Large vehicle, Ship, Tennis court, Basketball court, Storage tank, Soccer ball field, Roundabout, Harbor, Swimming pool, and Helicopter. Due to the large size of the DOTA dataset images, they cannot be directly used for training neural networks. Therefore, we divided the images into sub-images of size 608×608 pixels at intervals of 100 pixels. The sub-images were randomly extracted in an 8:1:1 ratio to create the training set, validation set, and testing set.

4.2. Evaluation Metrics

We adopt commonly used evaluation metrics, namely P (precision), R (recall), mAP (mean average precision), and $mAP_{0.5}$ (mean average precision at IOU = 0.5) in the experiments. Specifically, the expressions for P and R are defined as follows:

$$P = \frac{TP}{TP + FP} \tag{6}$$

$$R = \frac{TP}{TP + FN} \tag{7}$$

In this regard, TP represents the number of positive samples that were correctly identified, FP represents the number of negative samples that were identified as positive samples, and FN represents the number of positive samples that were identified as negative samples. Based on P and R, we can compute AP (average precision), mAP, and $mAP_{0.5}$ as follows:

$$\mathrm{AP} = \int_0^1 P(R) d\, R \tag{8}$$

$$\mathrm{mAP} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{AP}_i \,, \ \mathrm{IOU} = 0.5 : 0.05 : 0.95 \tag{9}$$

$$\mathrm{mAP}_{0.5} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{AP}_i \,, \ \mathrm{IOU} = 0.5 \tag{10}$$

where N represents the number of classifications of the targets.

4.3. Implementation Details

The implementation of CNTR-YOLO utilizes PyTorch (version v1.8.0) as the underlying framework, and the operating system used is Ubuntu 20.4. An NVIDIA RTX3060 GPU with 12 GB memory served as the platform for training and testing. During training, an SGD optimizer was used with the momentum and weight decay set to 0.937 and 0.01, respectively. A warmup strategy was employed to enhance the training process's stability. The learning rate gradually decreased at a rate of 0.01 for the first three epochs and continued training with 0.001. Moreover, the images were resized to 640 × 640 pixels, and considering the hardware limitations, the batch size was set to 2.

The other models, including Faster R-CNN, YOLOv4, YOLOv5m, YOLOv5l, and YOLOv5x, were tested and trained under the same settings as CNTR-YOLO, with images also resized to 640 × 640 pixels during training. Notably, we adopted the default settings of each model's referenced research articles concerning other parameters.

4.4. Experimental Results

In line with the implementation settings in Section 4.3, we evaluate CNTR-YOLO on P, R, mAP, $\mathrm{mAP}_{0.5}$, and Latency. To show the advantages of the proposed algorithm, we compare it with Faster R-CNN, YOLOv4, YOLOv5m, YOLOv5l, and YOLOv5x. We first present experimental results on the MAR20 dataset, and then, to demonstrate the robustness of the proposed algorithm, we also show experimental results on the DOTA dataset.

4.4.1. Experimental Results on the MAR20 Dataset

The overall comparison results are shown in Table 1. The comparison results of different categories are shown in Table 2.

Table 1. Comparison results of CNTR-YOLO and other algorithms on the MAR20 dataset.

Method	P (%)	R (%)	$\mathrm{mAP}_{0.5}$ (%)	mAP (%)	Latency (ms)
Faster R-CNN	77.3	73.6	82.7	57.1	83.6
YOLOv4	83.3	79.5	86.6	64.3	12.8
YOLOv5m	85.7	80.3	87.6	65.7	11.0
YOLOv5l	85.2	83.4	88.5	66.8	19.3
YOLOv5x	86.6	85.9	89.7	68.0	37.5
CNTR-YOLO	88.9	87.5	91.1	70.1	33.5

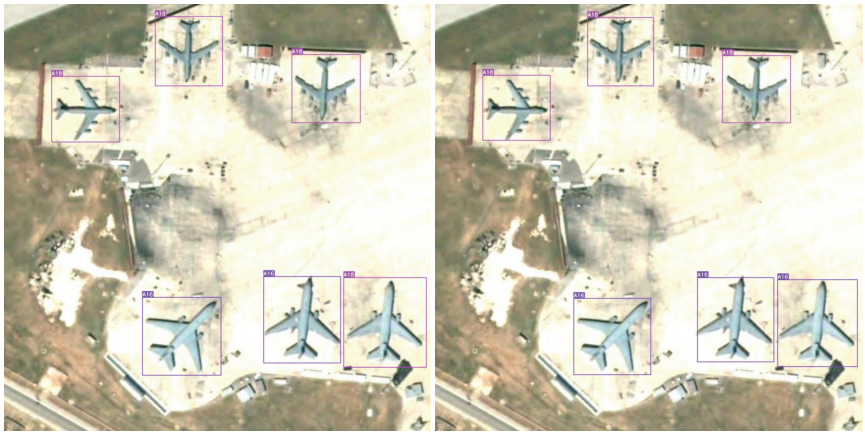
Table 1 presents the comparative results of six target detection algorithms using different metrics. CNTR-YOLO outperforms the others in terms of P, R, $\mathrm{mAP}_{0.5}$, and mAP. Specifically, CNTR-YOLO attains $\mathrm{mAP}_{0.5}$ and mAP scores of 91.1% and 70.1%, respectively, which are 1.4% and 2.1% higher than YOLOv5x, and 2.6% and 3.3% higher than YOLOv5l. In addition, CNTR-YOLO's mAP is 13.0% and 5.8% higher when compared against other non-YOLOv5 series algorithms, Faster R-CNN and YOLOv4, respectively. Notably, the proposed algorithm distinguishes different types of aircraft features with remarkable accuracy,

achieving a recall rate of 87.5%, which is 4.1% and 1.6% higher than YOLOv5l and YOLOv5x, respectively. This ability significantly reduces recognition errors compared to other algorithms. Despite a 14.2 ms higher inference time than YOLOv5l, CNTR-YOLO's improved detection performance still ensures that it is 4.0 ms faster than YOLOv5x.

Table 2. Comparison results of CNTR-YOLO and other algorithms on various categories of MAR20 dataset (mAP).

Class	Faster R-CNN	YOLOv4	YOLOv5m	YOLOv5l	YOLOv5x	CNTR-YOLO
A1	63.7	67.5	70.9	73.1	72.8	74.0
A2	67.5	75.3	75.7	77.9	77.6	80.9
A3	70.9	76.0	77.3	78.6	78.2	81.9
A4	66.8	71.9	73.4	76.3	75.2	75.3
A5	65.5	68.7	72.0	72.1	73.1	73.8
A6	55.8	68.5	70.8	71.6	69.4	74.7
A7	45.6	55.0	58.7	61.3	62.8	68.8
A8	59.7	67.0	67.8	70.5	70.4	72.2
A9	62.8	68.3	69.4	69.6	70.5	71.8
A10	50.1	59.3	59.2	62.5	64.6	66.5
A11	57.4	66.1	66.1	68.4	71.0	72.0
A12	40.1	43.2	45.2	48.0	48.7	46.7
A13	50.5	57.6	59.1	59.2	61.0	61.2
A14	26.6	32.0	31.6	33.0	37.4	42.7
A15	51.8	61.4	66.5	65.0	65.3	68.2
A16	58.9	64.1	67.2	63.3	69.2	74.5
A17	69.5	67.4	68.1	70.1	72.5	71.1
A18	65.9	71.5	71.4	73.0	72.7	74.0
A19	63.4	72.9	71.2	75.2	75.0	77.0
A20	63.5	71.6	72.5	73.1	73.7	74.8

Table 2 illustrates the mean average precision of the six methods across the twenty classifications in the MAR20 dataset. Overall, CNTR-YOLO outperforms the other five algorithms in most categories, with only three categories being inferior to YOLOv5l or YOLOv5, but the gaps are all within 2%. Notably, in category A14, where each method had the lowest mAP, CNTR-YOLO surpasses YOLOv5l and YOLOv5x by 9.7% and 5%, respectively. Additionally, CNTR-YOLO exhibits a significant performance advantage of 11.2% and 5.3%, respectively, over YOLOv5l and YOLOv5x in category A16. This gap is the largest among all categories. A comparison of the detection results between CNTR-YOLO and YOLOv5l on the same image is illustrated in Figure 11. CNTR-YOLO correctly identifies all instances, whereas YOLOv5l misidentifies an aircraft of A16 in the bottom right corner as belonging to A18. These two categories are visually similar from the perspective of remote sensing satellites (vertical direction), but CNTR-YOLO with its stronger detail discrimination ability can identify them correctly.



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

Figure 11. The detection results on one image of the test set of the MAR20 dataset: (**a**) detection result of YOLOv5 (**b**) detection result of CNTR-YOLO.

4.4.2. Experimental Results on the DOTA Dataset

Similarly, we show the general comparison results on the DOTA dataset in Table 3 and then show the comparison results on the specific categories in Table 4.

Table 3. Comparison results of CNTR-YOLO and other algorithms on the DOTA dataset.

Method	*P* (%)	*R* (%)	$mAP_{0.5}$ (%)	mAP (%)	Latency (ms)
Faster R-CNN	75.9	71.8	75.8	53.7	84.5
YOLOv4	80.2	75.7	80.3	59.3	13.3
YOLOv5m	82.9	78.8	81.8	60.6	11.6
YOLOv5l	82.8	80.3	82.4	61.2	20.0
YOLOv5x	83.3	82.1	83.6	62.0	38.2
CNTR-YOLO	85.1	84.3	85.2	63.7	34.1

Table 4. Comparison results of CNTR-YOLO and other algorithms on various categories of DOTA dataset (mAP).

Class	Faster R-CNN	YOLOv4	YOLOv5m	YOLOv5l	YOLOv5x	CNTR-YOLO
Plane	69.1	75.6	78.6	79.7	80.5	83.2
Basketball diamond	58.2	64.0	64.6	65.2	67.6	72.8
Bridge	33.6	37.3	38.8	39.1	39.5	40.1
Ground track field	55.5	63.7	64.7	65.0	67.1	66.9
Small vehicle	49.4	54.0	53.8	55.6	56.1	59.3
Large vehicle	66.9	72.8	73.8	74.9	76.8	76.5
Ship	63.7	70.1	69.8	72.5	73.0	74.1
Tennis court	84.9	89.2	91.2	90.6	92.0	91.9
Basketball court	70.5	75.1	76.5	77.0	78.4	79.1
Storage tank	57.6	64.4	66.9	67.7	68.1	70.4
Soccer ball field	24.3	28.2	28.5	28.9	28.3	30.1
Roundabout	46.2	52.9	55.8	55.3	56.1	57.5
Harbor	60.7	65.9	66.5	67.5	67.4	68.7
Swimming pool	48.9	55.3	56.6	57.1	57.3	58.9
Helicopter	16.3	20.7	22.3	22.1	22.6	25.5

From Table 3, it can be observed that on the DOTA dataset, the proposed algorithm yields superior $mAP_{0.5}$ and mAP compared to other algorithms. This indicates the robust-

ness of the proposed algorithm. When compared to YOLOv5l, CNTR-YOLO achieves 2.8% and 2.5% higher $mAP_{0.5}$ and mAP, respectively. When compared to YOLOv5x, CNTR-YOLO achieves 1.6% and 1.7% higher $mAP_{0.5}$ and mAP, respectively. In comparison to other algorithms, CNTR-YOLO outperforms them to a greater extent. The inference time results are very similar to those shown in Table 1, which is reasonable.

From Table 4, we can find that CNTR-YOLO outperforms other algorithms in most categories, which indicates that the proposed algorithm has a certain universality in the target detection of remote sensing images. Specifically, in the category of Plane, which represents the Aircraft considered in this paper, CNTR-YOLO yields mAP values of 83.2% that are 2.7% and 3.5% higher than those of YOLOv5x and YOLOv5l, respectively. This indicates that the proposed algorithm has superior performance in Aircraft detection compared to other algorithms on the DOTA dataset.

4.5. Ablation Study

The improvements of CNTR-YOLO include the substitution of a C3 with a Dense module, the application of the CBAM attention module, and the introduction of the C3CNTR module. These measures provide different levels of enhancements to YOLOv5l, which we will evaluate in this section. Although adding a small-scale detection head is common in YOLO-related object detection studies, such as TPH-YOLO, the approach is not utilized in this paper. The reason for this omission will be explained below. Furthermore, since C3CNTR is an improvement of C3TR, we will also inspect the enhancement effect of C3TR on the network (at the same position where C3CNTR is implemented). This assessment is essential to differentiate the performance variations between the two. The experimental results are displayed in Table 5, where the "tiny head" represents the small target detection head. It should be noted that to save table space, the suffixes "module" or "attention module" in the nouns of the tables are omitted in this paper.

Table 5. Results achieved by YOLOv5 combining different modules on the MAR20 dataset.

Method	*P* (%)	*R* (%)	$mAP_{0.5}$ (%)	mAP (%)
YOLOv5l	85.2	83.4	88.5	66.8
+tiny head	84.5	82.9	88.1	66.6
+Dense	87.0	84.8	89.8	68.0
+Dense+CBAM	87.5	85.3	89.9	68.2
+Dense+CBAM+C3TR	88.1	86.9	90.6	69.3
+Dense+CBAM+C3CNTR	88.9	87.5	91.1	70.1

Table 5 indicates that adding a small-scale target detection head reduces all metrics. This outcome is due to the majority of instances in the MAR20 dataset not being smaller than 32×32 pixels. Consequently, this approach is not employed in this paper. After incorporating the Dense module, all the performance metrics improved noticeably, and the mAP rose by 1.2% compared to YOLOv5l. Following the integration of the CBAM attention module, there were slight enhancements in all measures, resulting in a 0.2% increase in the mAP. In addition to these enhancements, the introduction of C3TR and C3CNTR produced different outcomes. While C3TR produced an increase of 1.1% in the mAP, C3CNTR resulted in a 1.9% increase, indicating that C3CNTR outperforms C3TR. Finally, after implementing all the improvements, CNTR-YOLO experiences a 3.3% enhancement in the mAP compared to YOLOv5l.

Regarding the use of attention mechanisms, several alternatives to CBAM were investigated, including Coordinate Attention (CA), Squeeze-and-Excitation Attention (SE), Normalization-based Attention (NAM), and Efficient Channel Attention (ECA); however, none of them achieved the anticipated outcome. After conducting experiments, we present the comparison results of the CBAM attention module and the aforementioned four alternatives on the MAR20 dataset in Table 6. It is worth noting that the experiments were based on the YOLOv5l+Dense module and represented by YOLOv5l*.

Table 6. Results achieved by YOLOv5 combining different modules on the MAR20 dataset; tiny head means small target detection head and Dense means Dense module.

Method	*P* (%)	*R* (%)	$mAP_{0.5}$ (%)	mAP (%)
YOLOv5l*	87.0	84.8	89.8	68.0
+CA	86.3	84.2	89.3	67.7
+SE	86.5	84.4	89.5	67.8
+NAM	87.0	84.6	89.7	68.0
+ECA	87.1	84.9	89.8	68.1
+CBAM	87.5	85.3	89.9	68.2

Table 6 reveals that CA and SE did not enhance the network's performance; instead, they caused a decline of 0.3% and 0.2% on mAP, respectively. NAM maintained the same level of performance, while ECA and CBAM elevated mAP by 0.1% and 0.2%, respectively.

5. Conclusions

In this paper, we propose the CNTR-YOLO algorithm for detecting aircraft targets in remote sensing images by improving the existing YOLOv5 algorithm. Our work includes the first attempt to combine a convolutional network and Transformer to design a new module in YOLOv5 as well as validates some improved measures to help YOLOv5 achieve better performance in aircraft detection. Specifically, our proposed C3CNTR module absorbs the local observation capability of ConvNext and the global analysis capability of Transformer, making a greater contribution to improving detection accuracy compared to the C3TR module that uses only Transformer. Next, during the feature extraction stage, the Dense module significantly improves the network's exploitation of features by utilizing multiple connections between convolutional layers, also avoiding the problem of gradient vanishing. Finally, we integrate the CBAM attention module to reduce interference from background information on the network, allowing the network to focus more on valuable areas and further improve the detection accuracy of the network. The mAP of the proposed CNTR-YOLO is 3.3% higher than YOLOv5l on the MAR20 dataset and exceeds other comparative methods, such as Faster R-CNN and YOLOv4. The results on the DOTA dataset show that the mAP of CNTR-YOLO reached 63.7%, also surpassing other compared methods. Particularly, for the specific category of Plane (which refers to aircraft in this paper), CNTR-YOLO achieved an mAP of 83.2%, which is 3.5% higher than YOLOv5l. This also reflects that our proposed algorithm has a certain robustness.

Author Contributions: Writing—Original draft preparation and software, F.Z. and Q.X.; Conceptualization and methodology, F.Z. and X.L.; Writing—Review and editing, F.Z. and H.D.; Resources, H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: We are grateful to the High-Performance Computing Center of Central South University for the assistance with the computations.

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

References

1. Blaschke, T. Object based image analysis for remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 2–16. [CrossRef]
2. Cheng, G.; Han, J. A Survey on Object Detection in Optical Remote Sensing Images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28. [CrossRef]
3. Zhang, L.B.; Zhang, Y.Y. Airport Detection and Aircraft Recognition Based on Two-Layer Saliency Model in High Spatial Resolution Remote-Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 1511–1524. [CrossRef]
4. Zuo, J.; Xu, G.; Fu, K.; Sun, X.; Sun, H. Aircraft Type Recognition Based on Segmentation with Deep Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 282–286. [CrossRef]

5. Zhao, Z.; Zheng, P.; Xu, S.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
6. Girshick, R.B.; Donahue, J.; Darrell, T.; 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.
7. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Berlin, Germany, 11–14 March 2015; pp. 1440–1448.
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
9. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
10. 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.
11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV 2016, Pt. I, Amsterdam, The Netherlands, 11–14 October 2016; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerland, 2016; pp. 21–37.
12. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
13. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
14. Liu, Q.; Xiang, X.; Wang, Y.; Luo, Z.; Fang, F. Aircraft detection in remote sensing image based on corner clustering and deep learning. *Eng. Appl. Artif. Intell.* **2019**, *87*, 103333. [CrossRef]
15. Shi, L.; Tang, Z.; Wang, T.; Xu, X.; Liu, J.; Zhang, J. Aircraft detection in remote sensing images based on deconvolution and position attention. *Int. J. Remote Sens.* **2021**, *42*, 4241–4260. [CrossRef]
16. Wu, Q.; Feng, D.; Cao, C.; Zeng, X.; Feng, Z.; Wu, J.; Huang, Z. Improved Mask R-CNN for Aircraft Detection in Remote Sensing Images. *Sensors* **2021**, *21*, 2618. [CrossRef] [PubMed]
17. Ji, F.; Ming, D.; Zeng, B.; Yu, J.; Qing, Y.; Du, T.; Zhang, X. Aircraft detection in high spatial resolution remote sensing images combining multi-angle features driven and majority voting CNN. *Remote Sens.* **2021**, *13*, 2207. [CrossRef]
18. Cao, C.; Wu, J.; Zeng, X.; Feng, Z.; Wang, T.; Yan, X.; Wu, Z.; Wu, Q.; Huang, Z. Research on Airplane and Ship Detection of Aerial Remote Sensing Images Based on Convolutional Neural Network. *Sensors* **2020**, *20*, 4696. [CrossRef] [PubMed]
19. Zhou, L.; Yan, H.; Shan, Y.; Zheng, C.; Liu, Y.; Zuo, X.; Qiao, B. Aircraft detection for remote sensing images based on deep convolutional neural networks. *J. Electr. Comput. Eng.* **2021**, *2021*, 1–16. [CrossRef]
20. Luo, S.; Yu, J.; Xi, Y.; Liao, X. Aircraft target detection in remote sensing images based on improved YOLOv5. *IEEE Access* **2022**, *10*, 5184–5192. [CrossRef]
21. Liu, Z.; Gao, Y.; Du, Q.; Chen, M.; Lv, W. YOLO-Extract: Improved YOLOv5 for Aircraft Object Detection in Remote Sensing Images. *IEEE Access* **2023**, *11*, 1742–1751. [CrossRef]
22. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767v1.
23. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934v1.
24. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
25. 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.
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst* **2017**, *30*, 1–11.
27. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
28. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
29. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
30. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
31. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. 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.
32. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2818–2826.

33. Woo, S.; Park, J.; Lee, J.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
34. Yu, W.Q.; Cheng, G.; Wang, M.J.; Yao, Y.Q.; Xie, X.X.; Yao, X.W.; Han, J.W. MAR20: A Benchmark for Military Aircraft Recognition in Remote Sensing Images. *Natl. Remote Sens. Bull.* **2022**, 1–11. [CrossRef]
35. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. Dota: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.

remote sensing

MDPI

Article

Enhanced CNN Classification Capability for Small Rice Disease Datasets Using Progressive WGAN-GP: Algorithms and Applications

Yang Lu [1,*], Xianpeng Tao [1], Nianyin Zeng [2], Jiaojiao Du [1] and Rou Shang [3]

[1] College of Information and Electrical Engineering, Heilongjiang Bayi Agricultural University, Daqing 163319, China
[2] Department of Instrumental and Electrical Engineering, Xiamen University, Xiamen 361005, China
[3] College of Electrical Engineering and Information, Northeast Petroleum University, Daqing 163318, China
* Correspondence: luyanga@sina.com

Abstract: An enhancement generator model with a progressive Wasserstein generative adversarial network and gradient penalized (PWGAN-GP) is proposed to solve the problem of low recognition accuracy caused by the lack of rice disease image samples in training CNNs. First, the generator model uses the progressive training method to improve the resolution of the generated samples step by step to reduce the difficulty of training. Second, to measure the similarity distance accurately between samples, a loss function is added to the discriminator that makes the generated samples more stable and realistic. Finally, the enhanced image datasets of three rice diseases are used for the training and testing of typical CNN models. The experimental results show that the proposed PWGAN-GP has the lowest FID score of 67.12 compared with WGAN, DCGAN, and WGAN-GP. In training VGG-16, GoogLeNet, and ResNet-50 with PWGAN-GP using generated samples, the accuracy increased by 10.44%, 12.38%, and 13.19%, respectively. PWGAN-GP increased by 4.29%, 4.61%, and 3.96%, respectively, for three CNN models over the traditional image data augmentation (TIDA) method. Through comparative analysis, the best model for identifying rice disease is ResNet-50 with PWGAN-GP in X2 enhancement intensity, and the average accuracy achieved was 98.14%. These results proved that the PWGAN-GP method could effectively improve the classification ability of CNNs.

Keywords: image data augmentation; small sample; progressive WGAN-GP; rice disease; CNN

Citation: Lu, Y.; Tao, X.; Zeng, N.; Du, J.; Shang, R. Enhanced CNN Classification Capability for Small Rice Disease Datasets Using Progressive WGAN-GP: Algorithms and Applications. *Remote Sens.* **2023**, *15*, 1789. https://doi.org/10.3390/rs15071789

Academic Editors: Maoguo Gong, Kai Qin, Yue Wu and Qiguang Miao

Received: 25 February 2023
Revised: 25 March 2023
Accepted: 25 March 2023
Published: 27 March 2023

1. Introduction

Rice is one of the most important food crops worldwide, especially in Asian countries, where it plays a crucial role in diets. According to statistics, more than 3 billion people rely on rice as their primary source of food, and rice production accounts for nearly 20% of the world's total grain output. Additionally, rice is a major export commodity for many countries and regions and has significant impacts on local economies and trade [1]. Rice disease is one of the main factors affecting the high quality, efficiency, and yield of rice, so the recognition of rice diseases is an important method to protect food security.

The traditional method of rice disease recognition relies on visual observation and monitoring by plant protection specialists. However, this method requires experienced rice specialists, and long periods of monitoring are costly on large farms. The shortage of rice specialists, especially in developing countries, prevents effective and timely rice disease control.

With the development of artificial intelligence technology, researchers at home and abroad have successfully applied machine learning methods to the automatic recognition and identification of crop diseases. For example, image processing-based techniques have been used for rice disease detection and recognition with high recognition accuracy [2,3]

involving support vectors [4,5], *k*-Nearest Neighbor [6], and decision trees [7]. Nevertheless, these methods have disadvantages, such as difficulties in implementation for large-scale training samples and solving multiple classification issues and sensitivity to the selection of parameters, which hinder the further improvement of the recognition effect. Significant breakthroughs in deep learning have been achieved with better results in recent years, which include convolutional neural (CNNs) networks [8,9] and migration learning [4,10] for image recognition.

Recently, deep learning has become a key technology for big data intelligence [11] and has been successfully applied to tasks of plant disease identification and classification. Compared with classical machine learning methods, deep learning has a more complex model structure with more powerful feature extraction capabilities. In [12], depthwise separable convolution was proposed for crop disease detection. Experimentally tested on a subset of the PlantVillage dataset, Reduced MobileNet achieved a classification accuracy of 98.34%, with a lower number of parameters than VGG and MobileNet. In [13], aiming at low power consumption and low performance of small devices, a depth-wise separable convolution (DSC)-based PLD (DSCPLD) recognition framework was proposed, which was tested on rice disease datasets, and the accuracy of using S-modified MobileNet and F-modified MobilNet reached 98.53% and 95.53%, respectively. In [4], the model for the classification of rice leaf disease images by ResNet-50 combined with the SVM method achieved an F1 score of 98.38%. In [14], to improve the accuracy of existing rice disease diagnosis, VGG-16 and GooLeNet models were used to train on a dataset of three painless species of diseases, and the experimental results showed that the average classification accuracies of VGG-16 and GoogLeNet were 92.24% and 91.28%, respectively. In [15], the authors constructed a novel rice blast recognition method based on CNN to identify 90% of diseased leaves and 86% of healthy leaves, respectively. Although the above methods achieve accurate recognition of rice diseases, deep learning techniques need to include large datasets that satisfy various criteria to obtain better recognition results. Note that using limited image datasets for training can lead to the overfitting of model training [16]. That is to say, training dataset size has a large impact on deep learning-based disease recognition methods, and their performance will be severely degraded in the case of small samples, uneven data distribution, etc. [17,18].

A strategy to solve the data shortage is to convert the original data to generate artificial data, which is usually called data augmentation. Data augmentation is achieved by executing geometric transformations, noise addition, interpolation, color transformation, and other operations on the original data. Common structures in convolutional neural networks include pooling layers, strided convolutions, and downsampling layers. When the position of the input image changes, the output tensor may change drastically. Therefore, convolutional neural networks may misclassify images that have undergone image processing transformations. This type of transformation can be used to enhance small sample image datasets. However, this data augmentation method does not increase the diversity of image features in the original dataset but only exploits the design flaw of convolutional neural networks [19]. Methods based on deep learning provide an effective and powerful way to learn the implicit representation of data distribution. Inspired by the zero-sum game in game theory, the Generative Adversarial Networks (GAN) model has been proposed in [20], which can learn how to approach the true distribution of data and has powerful capabilities in image generation. The original GAN suffers from the problems of difficult convergence, training, and control of the model. To deal with these problems, the Wasserstein Generative Adversarial Network (WGAN) has been proposed in [21] to solve the difficulty of training the original GAN. WGAN training is more stable and theoretically solves the pattern collapse and the gradient disappearance. Whereas WGAN causes issues such as gradient explosion when generating data due to direct weight cropping, which makes the model training unstable. Wasserstein Generative Adversarial Network with Gradient Penalized (WGAN-GP) was developed in [22], a generative adversarial network

that controls the gradient by gradient penalty to settle the matters of gradient explosion and pattern collapse.

At present, GAN has been employed effectively in the field of data enhancement. A method has been put forward in [23] based on deep learning for tomato disease diagnostics that uses the conditional generative adversarial network (CGAN) to produce synthetic images of tomato plant leaves. The recognition accuracy of this method in the classification of tomato leaf images into 5, 7, and 10 categories is 99.51%, 98.65%, and 97.11%, respectively. An infrared image-generation approach was designed in [24] depending on CGAN. This method can generate high-quality and reliable infrared image data. In [25], a model combining CycleGAN and U-net has been constructed and applied to a small dataset of tomato plant disease images. The results show that the model is better than the original CycleGAN. A fault recognition mechanism was presented in [26] for bearing small samples based on InfoGAN and CNN. The extracted time-frequency image features are input into InfoGAN for training to expand the data. Tested on the CWRU dataset, the results show that this method is better than other algorithms and models. In [27] a strategy was raised relying on a WGAN combined with a DNN. The cancer image is expanded by GAN to improve the classification accuracy and generalization of DNN. The results show that the classification accuracy of DNN using WGAN is the highest in comparison with other methods. CycleGAN has been put to use in [28] to retreat the CT segmentation task domain dataset for enhancement. The results display that the Dice score on the kidney increases from 0.09 to 0.66, and the effect is significant, while the improvement is small on the liver and spleen. However, WGAN-GP is still not effective at generating high-resolution images. Therefore, Tero Karras proposed Progressive GAN (ProGAN) in [29], a growing GAN-derived model, which generates very low-resolution images first, and then gradually increases the generated resolution during training to generate high-resolution images stably. In [30], a Dual GAN was proposed for generating high-resolution images of rice disease, which is used in the field of data enhancement. Dual GAN uses WGAN-GP to generate rice disease images, and Optimized-Real-ESRGAN is used to improve image resolution. The experimental results show that the accuracy of ResNet-18 and VGG-11 is improved by 4.57% and 4.1%, respectively. In [31], a novel neural network-based hybrid model (GCL) is proposed. CGL includes GAN for data enhancement, CNN for feature extraction, and LSTM for rice disease image classification. The experimental results show that the proposed method can achieve 97% accuracy for disease classification. In [32], a new convolutional neural network was proposed for the identification of three rice leaf diseases, using a GAN-based technique to augment the dataset. The experimental results showed that the proposed method achieved an accuracy of 98.23%. The above studies can show that GAN application is effective for data enhancement in small sample datasets, but the resolution and stability of the current generation are yet to be improved.

To alleviate the lack of image data on rice diseases, we introduce a Progressive WGAN-GP, which is based on the WGAN-GP model and combines a progressive training method. This model is applied to rice disease image data augmentation to increase the accuracy of the recognition model in small-sample datasets. By analyzing the three diseases in the collected dataset as well as the open-source dataset, the experimental results show that the method has good robustness and generalization ability and has a fine recognition effect under small sample conditions. The main contributions of this paper are twofold. (1) The progressive training method is introduced into the WGAN-GP model. In the field of rice disease image generation, the generation performs better than WGAN-GP, WGAN, and Deep convolutional GAN (DCGAN). (2) The experimental results show that the PWGAN-GP method can not only generate high-quality images of rice diseases but also apply the generated images to the CNNs training by blending the dataset with real images, which can improve the performance of CNNs, and obtain a higher recognition accuracy than other methods.

The remainder of this paper is organized as follows. In Section 2, we describe the source and the pre-processing of the data. Section 3 presents the theory related to PWGAN-

GP. Section 4 describes the experimental setup of the PWGAN-GP for the application problem of rice disease image generation as well as recognition. Section 5 analyzes the experimental data of image generation and the comparison with other methods. Conclusions are given in Section 6.

2. Dataset

The image dataset used in this paper is shown in Figure 1. The rice disease image (I) is obtained from an experimental farm field at Heilongjiang Bayi Agricultural University. The device used to capture these rice images is a Redmi K30 Pro phone. The image dataset includes rice leaf blast, rice leaf blight, rice leaf brown spot, and healthy rice leaf to increase the diversity of rice disease image samples that are captured separately under different cycles of rice growth, weather conditions, and lighting conditions. The rice disease image datasets (II) [33], (III) [34], and (IV) [35] are from open-source databases available on the web. Database (II) contains 3355 images with 4 categories and an image resolution of 2798 × 2798 pixels; database (III) contains 120 images with 3 categories and an image resolution of 3081 × 897 pixels, and database (IV) contains 2800 images with 5 categories and an image resolution of 256 × 256 pixels. Since the open-source database contains a variety of categories of images, this experiment eliminates the images whose categories are not consistent with the research direction of this paper.

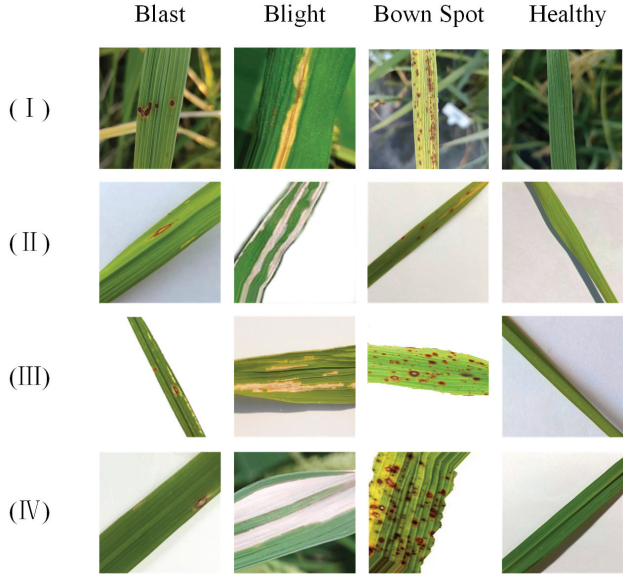


Figure 1. Images of rice diseases collected.

The dataset for this experiment consists of four sources, each with a high resolution from different sources, and in addition, there are differences in the methods of acquisition, which result in a non-uniform style of the dataset. Therefore, data pre-processing of the dataset is required. Duplicate, blurred, and images with insignificant disease characteristics are removed from the dataset. The number of categories in the pre-processed image dataset is shown in Table 1.

Table 1. Details of the rice leaf disease dataset.

Categories	Numbers
Blast	1654
Brown Spot	1570
Blight	1396
Healthy	2563

The workflow of the process of data segmentation and augmentation is shown in Figure 2. We randomly shuffled the order of the original dataset and split 80% of the image samples as the training set for data augmentation and image recognition, while the remaining 20% of the image samples served as the test set for an independent performance evaluation of the data-augmented image recognition. It is important to note that the data in the test set do not participate in the data augmentation phase in order to ensure the fairness of the test. The detailed numbers of training and test sets are shown in Table 2.

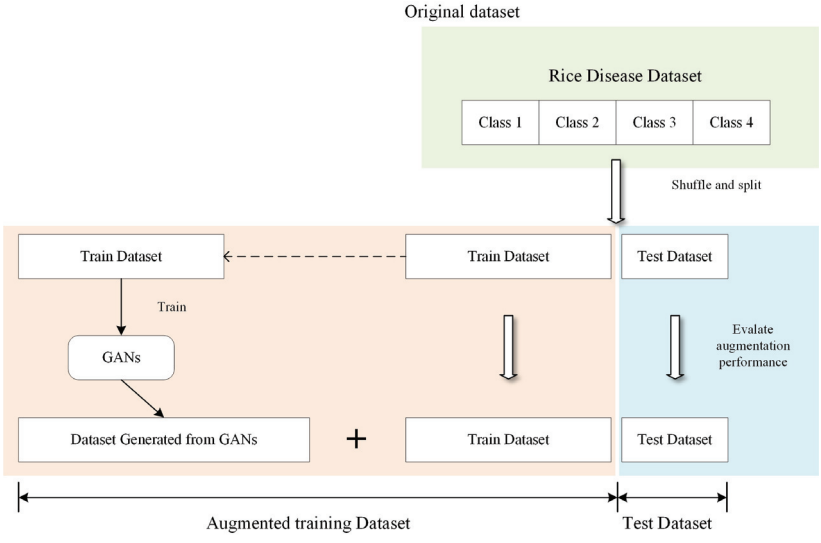


Figure 2. Dataset split strategy.

Table 2. Details of the rice leaf disease train and test dataset.

Categories	Train Dataset	Test Dataset
Blast	1323	331
Brown Spot	1256	314
Blight	1116	280
Healthy	2050	513

3. Methodology

3.1. GAN

GANs can be trained to generate high-quality images by learning the data distribution from the training set. GANs consist of two parts, one is the generator (G), and the other is the discriminator (D). The generator accepts the noise vector and generates samples. Then generates samples and real samples together to input into the discriminator, which needs to distinguish the real samples from the generated samples accurately. In the process of confrontation between the two models, the generated samples will be more realistic. At the same time, the discriminator's discriminatory ability will be enhanced. The generator and

discriminator will eventually game each other to reach the state of Nash equilibrium [20]. Because the samples generated by the GAN belong to the same labeled class as the original samples, they can be used for image dataset expansion. The objective function of the GAN is shown in Equation (1).

$$\min_G \max_D V(D,G) = E_{x\sim p_{data(x)}}[log(D(x))] + E_{z\sim p_\theta(z)}[log(1-D(G(z)))] \quad (1)$$

where $p_{data(x)}$ is the probability distribution of the real image and $\theta(z)$ is the input noise distribution of G. G and D fight against each other, with G continuously improving its ability to capture the true sample distribution and generate higher-quality images, and D improving its ability to discriminate the generated images. The original GAN has been shown to provide a more realistic output compared to other generative image algorithms.

However, there are three major problems with the original GAN as follows: (1) the loss function values of the generative and discriminant models are unstable during training, which indirectly leads to the instability of the generated images; (2) the original GAN architecture is prone to pattern collapse, where the generative model finds a limited range of samples from the original data that may result in the discriminator not being able to continue being effectively trained. In addition, the images generated by the generator lack diversity; and (3) adjusting the hyperparameters of the traditional GAN makes it very difficult to make the model converge.

3.2. WGAN

In [21], the theory related to Jensen-Shannon has been analyzed, which concludes that it is not reasonable to use Jensen-Shannon to measure the distance of disjoint parts between distributions. To improve the quality of the images generated by GAN, instead of using Jensen-Shannon, it is pointed out to use the Wasserstein distance measure as the distance between the generated data and the real data distribution. The definition of the Wasserstein distance is shown in Equation (2).

$$W(P_r,P_g) = \inf_{\gamma\in\prod(P_r,P_g)} E_{(x,y)\sim\gamma}[\|x-y\|] \quad (2)$$

where P_r is the distribution of the real data, P_g is the distribution of the generated data, and $\gamma \in \prod(P_r,P_g)$ is the joint distribution of P_r and P_g. The loss function of WGAN is shown in Equation (3).

$$L(D) = E_{z\sim P_z}[f_w(G(z))] - E_{x\sim P_x}[f_w(x)] \quad (3)$$

where z is the input noise and x is the real input image. $G(z)$ is the image generated by the received noise as the input of the generator. $E_{z\sim P_z}$ describes the probability distribution of the noise, $E_{x\sim P_r}$ denotes the probability distribution of the real image. f_w is the discriminator neural network containing parameter w in WGAN. The discriminator uses gradient clipping (weight clipping) so that the discriminator satisfies the condition of the Lipschitz constraint and restricts parameters w of the neural network f_w to be in a certain range $[-c,c]$.

The discriminator of WGAN does not directly distinguish between the generated sample and the real sample but measures the difference by calculating the Wasserstein distance. Therefore, as the value of the loss function decreases, the Wasserstein distance between the real sample and the generated sample approaches zero, meaning that the generated sample is closer to the real sample distribution. However, the use of gradient clipping in the WGAN may cause the weights to converge to the two extremes of the clipping range, leading to gradient explosion, gradient disappearance, an unreasonable generation along with the samples, and other side effects, as shown in Figure 3 [22].

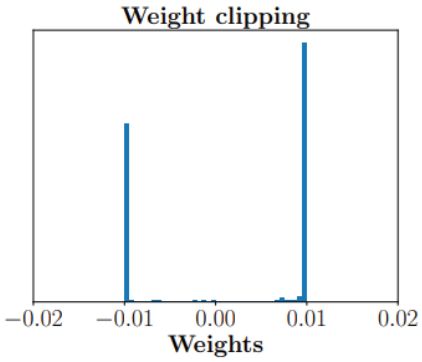


Figure 3. Weight clipping.

3.3. WGAN-GP

The WGAN-GP model has been proposed to solve this problem by allowing the discriminator to learn smoother decision boundaries through gradient penalty [22], as shown in Figure 4, and the gradient penalty implemented by WGAN-GP can satisfy the Lipschitz constraint. The loss function of WGAN-GP is shown in Equation (4).

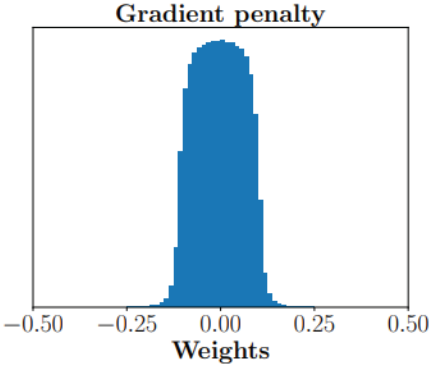


Figure 4. Gradient penalty.

$$L(D) = \mathop{\mathbb{E}}_{\tilde{x}\sim P_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim P_r}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{4}$$

where $\mathop{\mathbb{E}}_{\tilde{x}\sim P_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim P_r}[D(x)]$ is the loss function of WGAN, $\tilde{x} \sim P_g$ is the sampling of the generated data, and $x \sim P_r$ is the sampling of the real data. $\lambda \mathop{\mathbb{E}}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]$ is the gradient penalty term, $\hat{x}$ is the random noise therein, $\hat{x} \leftarrow \varepsilon x + (1-\varepsilon)\tilde{x}$ with random numbers $\varepsilon \sim U[0,1]$.

3.4. Progressive Training

In the traditional training GAN model, the structure of the generative and discriminative models is kept constant, and the resolution of the target images generated by the model is fixed. Due to the 'zero-sum game' characteristic of GAN, it is very difficult to train the model, and increasing the resolution of the generated images will further increase the training difficulty. In [29], a progressive training approach was proposed the key is to gradually increase the structure of the generative and discriminative models, starting

from low resolution, and after the training is stable, adding new layers to the generative and discriminative models, these layers will gradually model the details of the image. This both speeds up the training and stabilizes it greatly, resulting in clear and high-quality generated images.

3.5. Residual Block

The depth of a neural network has a large impact on the performance of the model, and as the depth increases, the model usually has better performance. However, as the network deepens, it is prone to the accuracy rising to a peak and then falling, a problem often called gradient degradation. In [36], the ResNet was proposed, and the key structure of the model is the residual block. The residual block makes features passing features, allowing subsequent network layers to pass less influence and uses all-equal mapping to pass inputs directly to outputs, ensuring the stable performance of the network. The structure of the residual module is shown in Figure 5.

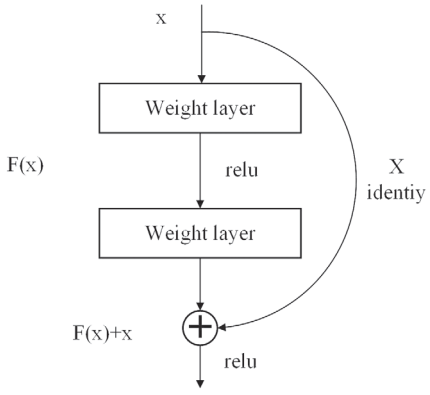


Figure 5. Residual block.

3.6. Progressive WGAN-GP

The Progressive WGAN-GP (PWGAN-GP) model consists of two parts: the generator and the discriminator. The generator consists of a residual block, an upsampling layer, and a LeakyReLU activation layer. The residual block of the generation model is responsible for generating image features, and the upsampling layer is responsible for scaling up the image size. The discriminant model consists of a residual module and a downsampling module. The residual module of the discriminator is responsible for extracting the image features, and the downsampling layer is responsible for reducing the image size. The loss function of WGAN-GP is used. In the training of the model, the residual block layer in the generator and discriminator increases step by step, and the size of the generated samples also increases. The training starts by generating a low-resolution 4×4 of the target image, and when the value of the loss function decreases to a stable state, it indicates that the training is completed at that stage. Next, the structure of the production model and the discriminant model is increased by one layer to continue the training. This is repeated to reach the preset target image resolution of 256×256. The training process is shown in Figure 6. A detailed description of this model is shown in Tables 3 and 4.

Table 3. Generator-related parameters of PWGAN-GP.

Layer Name	Activation Function	Output Tensor
Latent vector	-	512 × 1 × 1
Residual block	LeakyReLu	512 × 4 × 4
Upsample	-	512 × 8 × 8
Residual block	LeakyReLu	512 × 8 × 8
Upsample	-	512 × 16 × 16
Residual block	LeakyReLu	512 × 16 × 16
Upsample	-	128 × 32 × 32
Residual block	LeakyReLu	128 × 32 × 32
Upsample	-	64 × 64 × 64
Residual block	LeakyReLu	64 × 64 × 64
Upsample	-	32 × 128 × 128
Residual block	LeakyReLu	32 × 128 × 128
Upsample	-	16 × 256 × 256
Residual block	LeakyReLu	16 × 256 × 256
Conv 1 × 1	-	3 × 256 × 256

Table 4. Discriminator-related parameters of PWGAN-GP.

Layer Name	Activation Function	Output Tensor
Intput image	-	3 × 256 × 256
Conv 1 × 1	LeakyReLU	16 × 256 × 256
Residual block	LeakyReLU	32 × 256 × 256
Downsample	-	32 × 128 × 128
Residual block	LeakyReLU	64 × 128 × 128
Downsample	-	64 × 64 × 64
Residual block	LeakyReLU	128 × 64 × 64
Downsample	-	128 × 32 × 32
Residual block	LeakyReLU	256 × 32 × 32
Downsample	-	256 × 16 × 16
Residual block	LeakyReLU	512 × 16 × 16
Downsample	-	512 × 8 × 8
Residual block	LeakyReLU	512 × 8 × 8
Downsample	-	512 × 4 × 4
Avg pool, fc 1, softmax	-	1 × 1 × 1

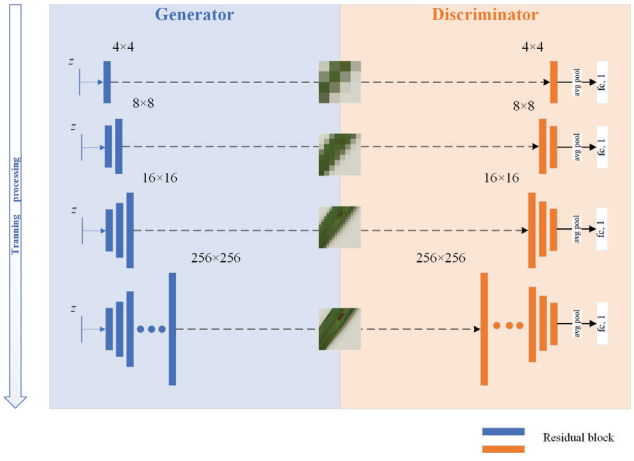


Figure 6. PWGAN-GP model.

Since a new layer is added at the end of each training phase, the new layer is still in the initialization state and cannot be directly added to the training; otherwise, it will affect the well-trained parameters as well. In this paper, the parameters of the old layer are fused into the parameter parameters of the new layer by a fusion mechanism. The formula is shown in (5).

$$Output = \alpha \times L_{new} + (1 - \alpha) \times L_{old} \tag{5}$$

$Output$ is the output of the new layer, α is the fusion coefficient factor, L_{new} is the parameter of the new layer, and L_{old} is the parameter of the old layer. The model multiplies the parameters of the old layer by $(\alpha - 1)$ plus the parameters of the new layer by α. The value increases from 0 to 1 one by one as the number of training increases. α takes values in the range $[0, 1]$. The structure is shown in Figure 7.

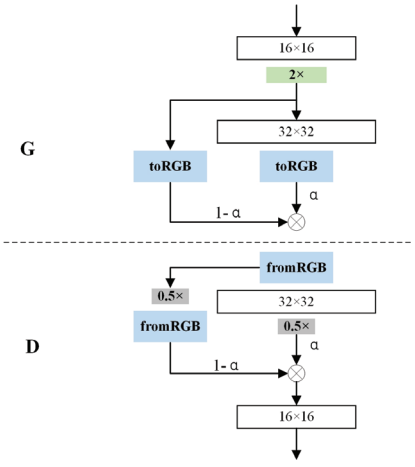


Figure 7. New layer fusion.

3.6.1. Residual Block

The residual block needs a convolutional layer to extract features of the input and differentiate the generated image and the real image. The convolutional layer applies the convolutional kernel and the activation function to calculate the feature map. The mathematical definition is shown in Equations (6) and (7).

$$y_j^l = f(z_j^l) \tag{6}$$

$$z_j^l = \sum_{i \in M_j} x_i^{l-1} \times k_{ij}^l + b_j^l \tag{7}$$

where z_j^i is the output of the feature map in the l-th layer, $f(\cdot)$ is the LeakyRelu activation function, z_j^l is the weight value of the j-th channel in the l-th layer, x_i^{l-1} is the feature map of the $(l-1)$-th layer, M_j is the subset of the input feature map, and k_{ij}^l is the convolution kernel matrix in layer l, $*$ means the Convolution operation, b_j^i means the bias term[37].

This paper uses a residual block with two layers of the same design. It includes a convolutional layer with a 4×4 convolutional kernel, the batch normalization layer, and the LeakyReLU activation layer. The structure is shown in Figure 8.

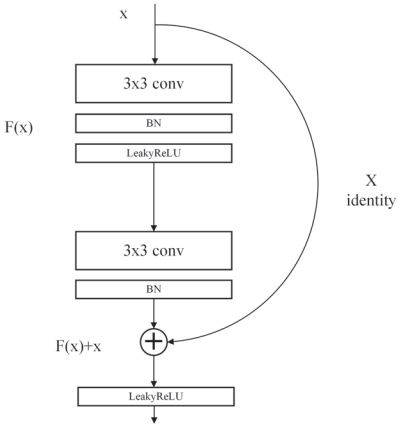


Figure 8. Two-layer residual block.

3.6.2. Upsampling

In this model, the Upsampling layer uses Transposed Convolution represented in deep learning as an inverse process of convolution. This approach can recover the image size and project the feature mapping to a higher dimensional space instead of recovering the original values. Transposed Convolution depends on the size of the convolution kernel and the size of the output. The formula for calculating the tensor size of outputs is shown in Equation (8).

$$o' = i' + (k - 1) - 2p \tag{8}$$

where o' represents the output size of the Transposed Convolution, i' denotes the size of the input Transposed Convolution, k depicts the size of the Transposed Convolution kernel, and p means the padding size when operating the tensor [38].

3.6.3. Batch Normalization Layer

Batch Normalization is a technique used in deep learning to improve the performance and stability of neural networks. The goal of Batch Normalization is to address the problem of internal covariate shift, which occurs when the distribution of the inputs to a layer changes during training. This can lead to slow convergence or even failure to converge. By normalizing the inputs to each layer, Batch Normalization can reduce the internal covariate shift and accelerate the training process [39]. The calculation formula of Batch Normalization is shown in Equations (9)–(12).

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i \tag{9}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} (\sum_{i=1}^{m} x_i - \mu_{\mathcal{B}})^2 \tag{10}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \tag{11}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \tag{12}$$

where m is the set batch size, x_i stands for the data of each batch, and $\mu_{\mathcal{B}}$ represents the mini-batch mean. $\sigma_{\mathcal{B}}^2$ means mini-batch variance. $\hat{x}_i$ indicates normalized. y_i reflects the output of Batch Normalizing Transform, γ is the equation coefficient, and β is the bias term [40].

3.6.4. LeakyReLU

The activation function is essentially the introduction of nonlinear factors into the neural network, through which the neural network can fit various curves. If the activation function is not used, the output of each layer will be a linear function of the input of the previous layer. By introducing a nonlinear function as the activation function, the output will be able to approximate any function. LeakyReLU is an activation function specifically designed to solve the Dead ReLU problem [41]. The mathematical description is shown in Equation (13).

$$LeakyReLU = \begin{cases} x, x > 0 \\ \alpha x, x \leq 0 \end{cases} \tag{13}$$

The *LeakyReLU* function adjusts the zero-gradient problem for negative values by giving a very small linear component of x to the negative input multiplied by 0.01, usually with a value of α of about 0.01. Its function range is negative infinity to positive infinity.

3.7. Traditional Image Data Augmentation

CNN is a powerful model for abstracting features from unstructured data, but they do not have image invariance because of the down-sampling operation that changes the image [42]. Then, the performance of neural networks can be improved by performing some transformations on the dataset to generate a large number of diverse samples to make the neural networks have good robustness. This is realized using data expansion and increasing the number of training sessions is necessary. For the network to obtain invariance to the affine transformation of the samples, the network is usually trained using the Traditional image data augmentation (TIDA) approach. We use rotation, translation, scaling, brightness adjustment, contrast adjustment, and adding noise to transform the images. The transformed images are used to perform data augmentation on the original dataset and are compared with GAN data augmentation methods, as shown in Figure 9.

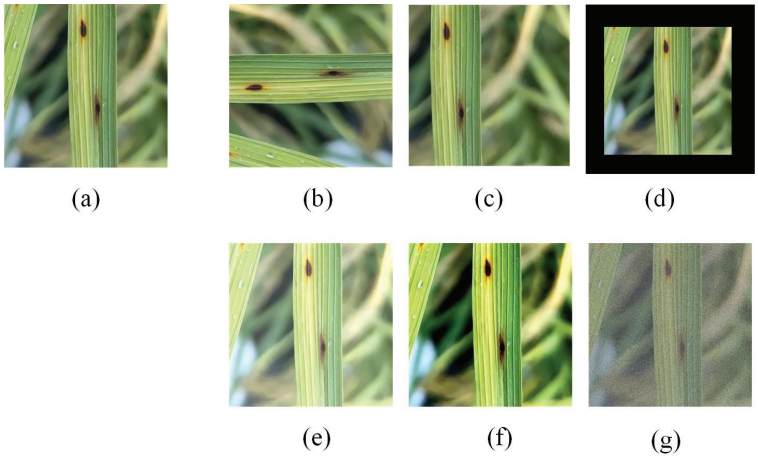


Figure 9. TIDA approach, where (**a**) is the original image, (**b**) rotation, (**c**) panning, (**d**) scaling, (**e**) brightness adjustment, (**f**) contrast adjustment, and (**g**) adding noise.

4. Experiment

In this paper, we validate the effectiveness of the generated data in two aspects as follows: (1) evaluating the quality of the generated data; and (2) assessing the impact of the generated data on the performance of the deep learning model.

4.1. Experimental Setup

The experimental environment is 15 vCPU Intel(R) Xeon(R) Platinum 8358P CPU @ 2.60 GHz processor, 32 G memory, RTX A5000 (24 GB) dual graphics card, Ubuntu 20.04, PyTorch 1.10.0, CUDA 11.3 deep learning platform. The proposed experimental framework is shown in Figure 10.

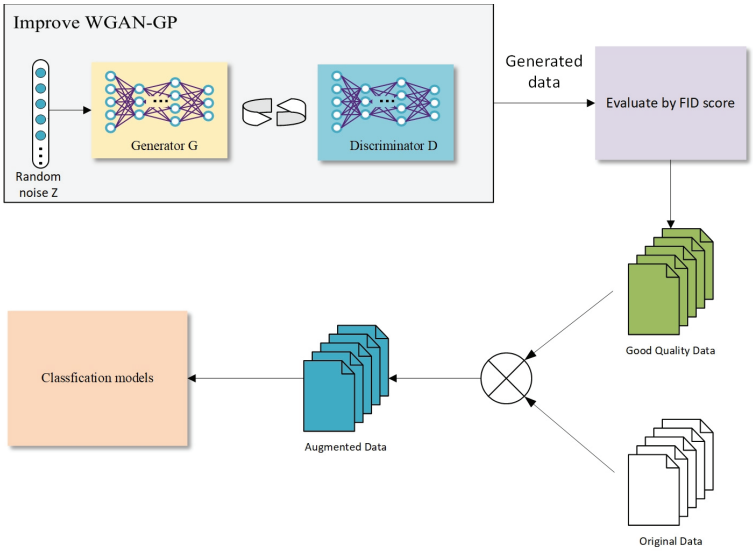


Figure 10. Flow chart of the experimental framework.

4.2. Evaluation Metrics

To verify that the PWGAN-GP network designed in this paper can perform the task of generating rice leaf disease images well, an experiment is set up to compare three classical generative adversarial models, i.e., WGAN, WGAN-GP, and DCGAN. The hyperparameters of the generative adversarial model are set to 20,000 epochs, the number of batches per batch is 128, and the learning rate is set to 0.0002. The Fréchet Inception Distance (FID) [43] metric is used to measure the similarity between the rice leaf disease images generated by the above models and the real images, and the lower the FID score means the two datasets have more similar distributions. The FID score is defined as shown in Equation (14).

$$FID = \|\mu_x + \mu_g\| + Tr(\Sigma_x + \Sigma_g - 2\sqrt{\Sigma_x \times \Sigma_g}) \tag{14}$$

where μ_x and Σ_x are the mean and covariance matrices of the 2048-dimensional feature vector set output by the real image collection in Inception-v3, respectively. μ_g and Σ_g are the mean and covariance matrices of the 2048-dimensional feature vector set output by the generated image collection in Inception Net-v3, respectively. Tr denotes the trace of the matrix.

4.3. Training Process

A random noise z is used as the input of the PWGAN-GP network, and the network is set to train for 20,000 epochs. To be able to monitor the training of the WGAN-GP network in a prompt manner and to evaluate the generation capability, the generated data are stored once every 200 epochs during the training process. Then, the FID score is used to measure the generated samples. The generator of PWGAN-GP generates a large number of high-quality generation samples, which are merged with the original samples for data augmentation. To verify the effectiveness of the data samples generated by the proposed framework, we test the data-augmented samples with the classical classification model.

To verify the effectiveness of PWGAN-GP for rice disease image data augmentation, the original data are randomly divided into a training set and a test set at a ratio of 8:2. The training set is used to train the WGAN-GP model and measure the generation quality of the model by the FID score. The generator of PWGAN-GP is applied to generate an image dataset with a similar distribution to the real image sample. Then, the generated image samples and original training set are mixed to enhance the performance of the CNN model.

4.4. Performance of the Data Augmentation Model

To verify the effectiveness of the rice leaf disease images generated by GANs on the original image dataset enhancement, classical CNN models such as VGG-16 [44], GoogLeNet [45], and ResNet-50 [46] are selected to test the enhanced dataset with accuracy as the main evaluation index of the test. In addition, three enhancement levels (i.e., X1, X2, and X3) are set to analyze the effect on the ratio of the original data to the generated data, where X0 is the original data, X1, X2, and X3 (1:10, 1:20, and 1:30) indicate 10-fold, 20-fold, 30-fold augmentations based on the original data, respectively.

5. Results and Discussion

In this section, experimental results on the quality of the generated data shown can demonstrate the difference between the samples generated by PWGAN-GP and other generative adversarial models and the impact of the enhancement ratio on the original data samples on the absorption of the neural network classification. Finally, the advantages of PWGAN-GP compared to TIDA methods are also discussed, and validation of the CNN model after data augmentation is tested.

5.1. Generating Image Quality

As the data in Table 5 show, the average FID score of the WGAN is the highest, which indicates that the WGAN has the worst effect on the quality of the generated rice disease image dataset. The FID score of DCGAN decreases by 31.69 compared to the WGAN and is 20.66 higher compared to WGAN-GP, so the image generation effect of DCGAN is better than the WGAN and weaker than WGAN-GP. Dual GAN's FID score is close to that of WGAN-GP. The FID score of PWGAN-GP is the smallest among the comparison models, so the generation effect is also the best.

Table 5. Generation Result Evaluation of GANs by FID score.

Method	Blast	Brown Spot	Blight	Healthy	FID Score Average
WGAN	118.42	133.71	137.51	131.84	130.37
DCGAN	95.37	107.26	101.68	90.39	98.68
Dual GAN	70.13	86.78	92.24	64.20	78.34
WGAN-GP	75.18	84.96	79.33	72.61	78.02
PWGAN-GP	62.11	71.24	74.38	60.73	67.12

The details of the rice leaf disease-generated image are shown in Figure 11. It can be seen that the image generated by the original GAN has artifacts, the overall image is blurred, the edges of the leaf in the complex background are not clear, and, most importantly, the detail characteristics of the disease spots are seriously lost. Although the image clarity of the samples generated by Dual GAN is better than that of WGAN-GP, excessive processing of leaf and disease textures leads to excessive detail loss, so the generation effect is not improved. The details of the leaf and disease spots of PWGAN-GP-generated images are substantially improved and close to the real sample, but there are problems of distortions and local blurring. The training results of PWGAN-GP are shown in Figure 12. The image generated by PWGAN-GP has a stable structure with clear edges, most of the details of the lesions are preserved, and the overall sharpness of the image is further improved.

Therefore, the PWGAN-GP-generated rice leaf disease images are the best among the selected GAN models.

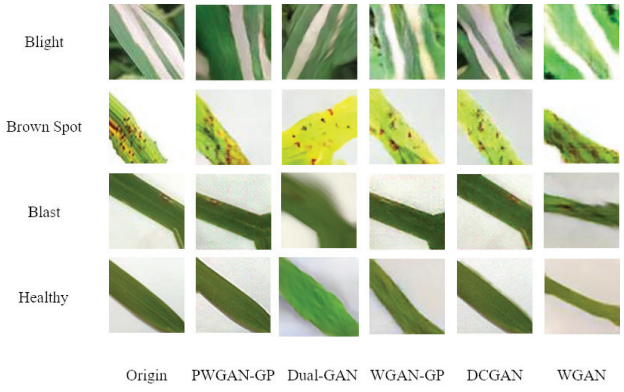


Figure 11. Comparison of generated samples.

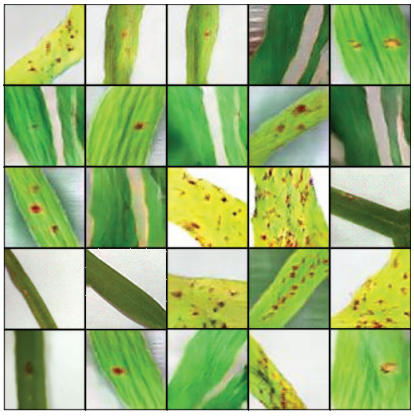


Figure 12. The images generated by PWGAN-GP.

Although the GAN model has strong feature-learning capabilities, it requires a lot of computational power and a longer training time. The training time for PWGAN-GP, WGAN-GP, DCGAN, and WGAN is shown in Table 6.

Table 6. Time spent on model training.

Method	Training Time (h)
WGAN	45
DCGAN	52
WGAN-GP	59
PWGAN-GP	88
Dual GAN	97

PWGAN-GP training requires a certain number of samples, and when the training set is too small, PWGAN-GP training will be affected, and it cannot produce effective images. The training dataset is reduced to 20%, 40%, 60%, and 80% for testing, and the experimental results are shown in Figure 13. As the dataset is reduced, the generated samples are distorted, blurred, and color confused.

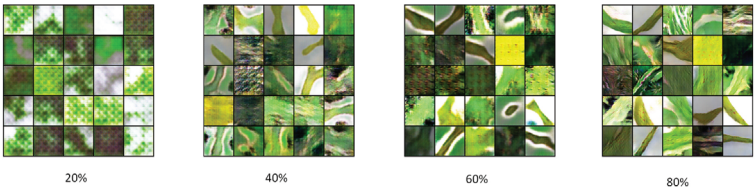


Figure 13. Effect of reducing the number of training sets on PWGAN-GP.

5.2. Performance of the Data Enhancement Model

The results of the tests using the VGG-16, GoogLeNet, and ResNet-50 models with different levels of enhancements to the original data are shown in Tables 7–9. The first row of each table shows the performance of the baseline model, and the next rows represent the accuracy values for enhancement levels X1, X2, and X3. The bolded numbers show the highest accuracy values for a single category in the test results. The last row of the table shows the maximum accuracy improvement compared with the benchmark model. The numerical units in the table are expressed using percentages. The experimental results display that the VGG-16, GoogLeNet, and ResNet-50 models show a significant increase in classification accuracy for different disease categories, including healthy leaves, after data augmentation. The data visualization is shown in Figure 14. Among them, ResNet-50 has the highest increased accuracy of 14.04%, 13.13%, 12.41%, and 12.18%, respectively, compared to the original data. In addition, the best enhancement intensity is X2 (1:20) for the three models, which has the best effect on the accuracy increase for the deep learning classification model.

Table 7. The effect of the strength of data enhancement on the accuracy of the VGG-16 model (%).

Level	Blast	Brown Spot	Blight	Healthy
X0	83.21	79.76	80.11	82.62
X1	88.48	85.81	87.83	**91.97**
X2	**93.77**	**89.28**	**91.35**	90.31
X3	90.52	88.31	89.7	88.27
Max. Improve	10.56	9.52	11.24	9.35

Table 8. The effect of the strength of data enhancement on the accuracy of the GoogLeNet model (%).

Level	Blast	Brown Spot	Blight	Healthy
X0	83.62	82.53	82.73	84.17
X1	94.84	94.03	93.84	94.16
X2	**96.26**	94.85	**94.91**	**95.37**
X3	95.53	**95.01**	94.69	95.21
Max. Improve	12.64	12.48	12.18	11.20

Table 9. The effect of the strength of data enhancement on the accuracy of the ResNet-50 model (%).

Level	Blast	Brown Spot	Blight	Healthy
X0	84.21	82.09	83.53	85.07
X1	96.77	94.74	95.48	96.81
X2	**98.25**	**95.22**	**95.94**	**97.19**
X3	97.63	94.44	94.23	96.98
Max. Improve	14.04	13.13	12.41	12.18

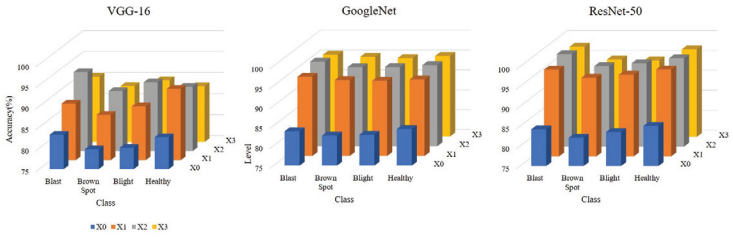


Figure 14. The effect of the level of data enhancement on the accuracy of neural network models.

The experimental results of the effects of different data enhancement methods on the training accuracy of neural networks are shown in Tables 10–12. The experiments of training neural network classification models using the original data directly with the enhanced dataset are compared between adopting the TIDA method and adopting the PWGAN-GP data enhancement method. The experimental results show that the TIDA method and the PWGAN-GP data augmentation method have a significant increase in the classification accuracy of VGG-16, GoogLeNet, and ResNet-50. The TIDA method increased by 7.24%, 8.52%, and 10.08%, respectively, over the situation without data augmentation in the average accuracy metrics of the three models. It can be shown that the data augmentation of the TIDA method can improve the recognition accuracy and generalization ability of the classical CNN models to some extent. PWGAN-GP increased by 10.44%, 12.38%, and 13.19%, respectively, over the situation without data augmentation. PWGAN-GP increased by 3.2%, 3.86%, and 3.11%, respectively, over the TIDA method. It can be seen that PWGAN-GP can significantly increase the accuracy and improve the generalization ability of the classical CNN model compared with the TIDA method. A visual analysis of the impact of data augmentation is shown in Figure 15 on the accuracy of the neural network model.

Table 10. Impact of data augmentation on the accuracy of the VGG-16 model (%).

Method	Blast	Brown Spot	Blight	Healthy	Avg.
Actual data	83.21	79.76	80.11	82.62	81.03
TIDA	88.15	88.04	88.71	89.16	88.27
PWGAN-GP	93.77	89.28	91.35	90.31	91.47

Table 11. Impact of data enhancement on the accuracy of the GoogLeNet model (%).

Method	Blast	Brown Spot	Blight	Healthy	Avg.
Actual data	83.62	82.53	82.73	84.17	82.96
TIDA	91.44	90.57	91.43	92.46	91.48
PWGAN-GP	96.26	94.85	94.91	95.37	95.34

Table 12. Impact of data enhancement on the accuracy of the ResNet-50 model (%).

Method	Blast	Brown Spot	Blight	Healthy	Avg.
Actual data	84.21	82.09	83.53	85.07	83.28
TIDA	93.12	93.31	93.18	93.83	93.36
PWGAN-GP	98.25	95.22	95.94	97.19	96.47

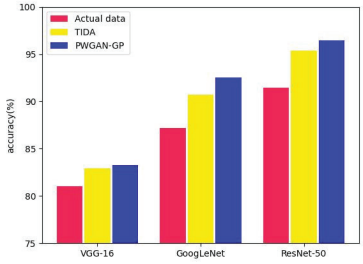


Figure 15. Impact of data enhancement on the accuracy of neural networks.

To obtain the best hyperparameters for ResNet-50 on rice disease identification, relevant validation experiments were conducted on learning rate, batch size, and optimizer. The hyperparameters of the experiment are shown in Table 13. The test results of hyperparameter selection experiments are shown in Figure 16. It can be seen that ResNet-50 performs best when the learning rate is 0.005, the batch size is 128, and the optimizer is RMSProp. The training under the optimal hyperparameter condition is shown in Figure 17. The accuracy of Resnet-50 is improved to 98.14%.

Table 13. Hyper-parameter details of ResNet-50.

Hyperparameter	Condition
learning rate	0.001, 0.005, 0.01, 0.05, 0.1
batch size	16, 32, 64, 128, 256
optimizer	SGD, Adam, RMSProp

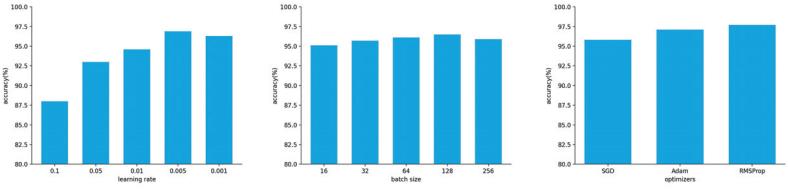


Figure 16. Hyperparameter optimization of ResNet-50.

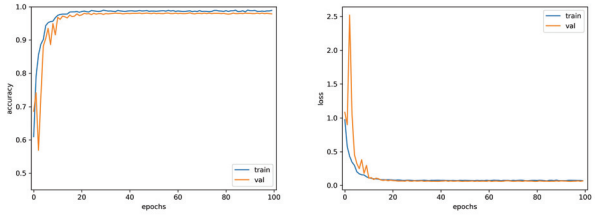


Figure 17. ResNet-50 training chart under optimal hyperparameter.

From Table 2, it can be seen that the test set is also imbalanced due to imbalanced datasets. An imbalanced test dataset may affect the test results of the model. Therefore, we adjusted the number of all categories in the test set to 280, manually simulated a balanced test set, and used the ResNet-50 with PWGAN-GP data augmentation to test. The experiment was repeated five times to find the average; its performance on balanced and imbalanced datasets is shown in Table 14 and Figure 18. The experimental results show that the performance of the enhanced ResNet-50 model on balanced and imbalanced datasets is close. Therefore, imbalanced test sets have little impact on test results.

Table 14. The influence of the imbalanced datasets on ResNet-50 testing.

Dataset Type	Average Accuracy (%)
Balanced dataset	98.04
Imbalanced dataset	98.33

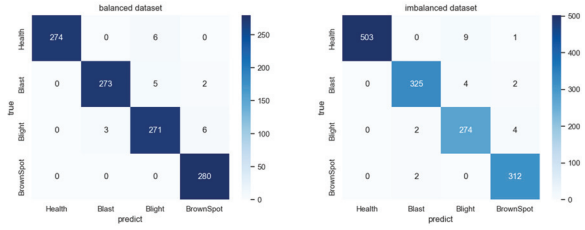


Figure 18. Confusion matrix for the effect of imbalanced test set on ResNet-50 test results.

Complex situations, such as overlapping disease features, exist in natural environments [47]. In order to test the recognition effect of the data augmentation model under complex disease feature conditions, we selected samples with complex backgrounds from the field-collected rice dataset as the test set, as shown in Figure 19. The number of test sets is shown in Table 15.

Figure 19. Datasets in complex environments.

Table 15. Details of the datasets in complex environments.

Categories	Numbers
Blast	56
Brown Spot	62
Blight	60
Healthy	60

From Table 16, ResNet-50 without data augmentation had a minimum accuracy of 81.55% in a complex background, indicating weak generalization of the model with insufficient data. The accuracy of ResNet-50 with TIDA is 94.84%, and the accuracy of ResNet-50 with PWGAN-GP is the highest, reaching 97.03%. The model is shown to have good generalization. Under the condition of overlapping features, the main convolutional outputs and feature maps of each layer during the inference of ResNet-50 are shown in Figure 20.

Table 16. ResNet-50 testing in the dataset of complex environments.

Model	Average Accuracy (%)
ResNet-50	81.55
TIDA+ResNet-50	94.84
PWGAN-GP+ResNet-50	97.03

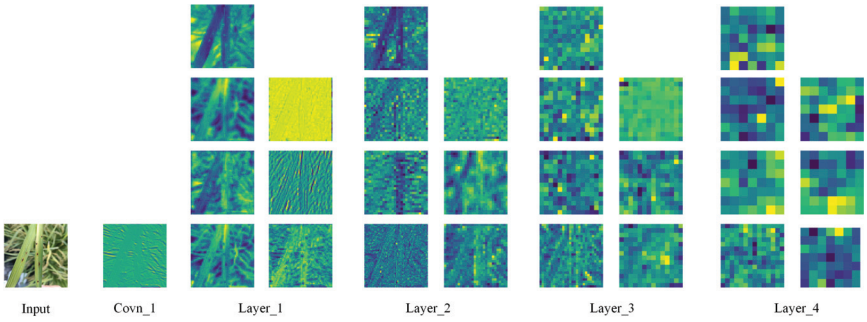


Figure 20. ResNet-50 forward propagation feature map.

6. Conclusions

To solve the problem of low accuracy caused by the lack of rice disease image datasets in training CNNs, PWGAN-GP is proposed to generate rice leaf disease images in this paper. First, we use the progressing training method to train the generator model and discriminator model, and a loss function is added to the discriminator model. It has been concluded that the PWGAN-GP network is the best to generate rice leaf disease images compared with WGAN, DCGAN, and WGAN-GP. Second, the experimental results show that the accuracy of VGG-16, GoogLeNet, and ResNet-50 using PWGAN-GP is 10.44%, 12.38%, and 13.19% higher than those without PWGAN-GP. Compared with a traditional image data augmentation method, the accuracy is increased by 3.2%, 3.86%, and 3.11%, respectively. The accuracy of CNNs can be maximized under the condition of X2 (1:20) enhancement intensity. Finally, under hyperparameter optimization, the ResNet-50 with PWGAN-GP achieved 98.14% for identifying three rice diseases. In addition, we also tested the performance of ResNet-50 in some scenarios, and the results were good. Therefore, it has been shown that PWGAN-GP has better image generation ability and improves the classification ability of CNNs.

At present, the model proposed in this paper also has the problem of long training time and slow convergence. In future work, we will solve these two problems by optimizing model parameters and combining deep learning with control theory [48–52].

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

Funding: This work was supported in part by the National Natural Science Foundation of China under Grants U21A2019, 61873058, 61933007, and 62373271, Heilongjiang Natural Science Foundation of China under Grant LH2020F042, the Scientific Research Starting Foundation for Post Doctor from Heilongjiang of China under Grant LBH-Q17134.

Data Availability Statement: The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Conflicts of Interest: The authors declare that there is no conflict of interests regarding the publication of this article.

References

1. Huang, S.; Wang, P.; Yamaji, N.; Ma, J.F. Plant Nutrition for Human Nutrition: Hints from Rice Research and Future Perspectives. *Mol. Plant* **2020**, *13*, 825–835. [CrossRef]
2. Gayathri Devi, T.; Neelamegam, P. Image processing based rice plant leaves diseases in Thanjavur, Tamilnadu. *Clust. Comput.* **2019**, *22*, 13415–13428. [CrossRef]
3. Chawathe, S.S. Rice disease detection by image analysis. In Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6–8 January 2020; pp. 0524–0530.

4. Sethy, P.K.; Barpanda, N.K.; Rath, A.K.; Behera, S.K. Deep feature based rice leaf disease identification using support vector machine. *Comput. Electron. Agric.* **2020**, *175*, 105527. [CrossRef]
5. Sulistyaningrum, D.; Rasyida, A.; Setiyono, B. Rice disease classification based on leaf image using multilevel Support Vector Machine (SVM). *J. Phys. Conf. Ser.* **2020**, *1490*, 012053. [CrossRef]
6. Adiyarta, K.; Zonyfar, C.; Fatimah, T. Identification of rice leaf disease based on rice leaf image features using the k-Nearest Neighbour (k-NN) technique. In Proceedings of the International Conference on IT, Communication and Technology for Better Life, ICT4BL, Bangkok, Thailand, 17–18 July 2019; pp. 160–165.
7. Mekha, P.; Teeyasuksaet, N. Image Classification of Rice Leaf Diseases Using Random Forest Algorithm. In Proceedings of the 2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering, Cha-am, Thailand, 3–6 March 2021; pp. 165–169.
8. Rasjava, A.R.; Sugiyarto, A.W.; Kurniasari, Y.; Ramadhan, S.Y. Detection of Rice Plants Diseases Using Convolutional Neural Network (CNN). In Proceedings of the International Conference on Science and Engineering, Male, Maldives, 14–16 January 2020; Volume 3, pp. 393–396.
9. Zhang, X.; Qiao, Y.; Meng, F.; Fan, C.; Zhang, M. Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access* **2018**, *6*, 30370–30377. [CrossRef]
10. Swasono, D.I.; Tjandrasa, H.; Fathicah, C. Classification of tobacco leaf pests using VGG16 transfer learning. In Proceedings of the 2019 12th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 18 July 2019; pp. 176–181.
11. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
12. Kc, K.; Yin, Z.; Wu, M.; Wu, Z. Depthwise Separable Convolution Architectures for Plant Disease Classification. *Comput. Electron. Agric.* **2019**, *165*, 104948. [CrossRef]
13. Hossain, S.M.M.; Deb, K.; Dhar, P.K.; Koshiba, T. Plant Leaf Disease Recognition Using Depth-Wise Separable Convolution-Based Models. *Symmetry* **2021**, *13*, 511. [CrossRef]
14. Yakkundimath, R.; Saunshi, G.; Anami, B.; Palaiah, S. Classification of Rice Diseases Using Convolutional Neural Network Models. *J. Inst. Eng. (India) Ser. B* **2022**, *103*, 1047–1059. [CrossRef]
15. Liang, W.j.; Zhang, H.; Zhang, G.f.; Cao, H.x. Rice blast disease recognition using a deep convolutional neural network. *Sci. Rep.* **2019**, *9*, 1–10. [CrossRef]
16. Belkin, M.; Ma, S.; Mandal, S. To understand deep learning we need to understand kernel learning. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 541–549.
17. D'souza, R.N.; Huang, P.Y.; Yeh, F.C. Structural analysis and optimization of convolutional neural networks with a small sample size. *Sci. Rep.* **2020**, *10*, 834. [CrossRef]
18. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 1–54. [CrossRef]
19. Zhang, R. Making Convolutional Networks Shift-Invariant Again. *arXiv* **2019**. [CrossRef]
20. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1–9.
21. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
22. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved Training of Wasserstein GANs. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
23. Abbas, A.; Jain, S.; Gour, M.; Vankudothu, S. Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Comput. Electron. Agric.* **2021**, *187*, 106279. [CrossRef]
24. Bing, L.; Yong, X.; Daqiao, Z. Infrared Image Generation Algorithm Based on Conditional Generation Adversarial Networks. *Acta Photonica Sin.* **2021**, *50*, 1110004.
25. Nazki, H.; Lee, J.; Yoon, S.; Park, D.S. Image-to-image translation with GAN for synthetic data augmentation in plant disease datasets. *Smart Media J.* **2019**, *8*, 46–57. [CrossRef]
26. Yang, Q.; Lu, J.G.; Tang, X.H.; Gu, X.; Sheng, X.J.; Yang, R.H. Bearing small sample fault diagnosis based on InfoGAN and CNN. *J. Ordnance Equip. Eng.* **2021**, *42*, 235–240.
27. Liu, Y.; Zhou, Y.; Liu, X.; Dong, F.; Wang, C.; Wang, Z. Wasserstein GAN-based small-sample augmentation for new-generation artificial intelligence: A case study of cancer-staging data in biology. *Engineering* **2019**, *5*, 156–163. [CrossRef]
28. Sandfort, V.; Yan, K.; Pickhardt, P.J.; Summers, R.M. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Sci. Rep.* **2019**, *9*, 16884. [CrossRef]
29. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.
30. Zhang, Z.; Gao, Q.; Liu, L.; He, Y. A High-Quality Rice Leaf Disease Image Data Augmentation Method Based on a Dual GAN. *IEEE Access* **2023**, *11*, 21176–21191. [CrossRef]
31. Lamba, S.; Baliyan, A.; Kukreja, V. A Novel GCL Hybrid Classification Model for Paddy Diseases. *Int. J. Inf. Technol.* **2023**, *15*, 1127–1136. [CrossRef] [PubMed]

32. Lamba, S.; Baliyan, A.; Kukreja, V. GAN Based Image Augmentation for Increased CNN Performance in Paddy Leaf Disease Classification. In Proceedings of the 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 28–29 April 2022; pp. 2054–2059. [CrossRef]
33. Shayan, R. Rice Leafs. 2019. Available online: https://www.kaggle.com/shayanriyaz/riceleafs (accessed on 24 February 2023).
34. Marsh. Rice Leaf Diseases Dataset. 2019. Available online: https://www.kaggle.com/vbookshelf/rice-leaf-diseases (accessed on 24 February 2023).
35. Rajeshbhattacharjee. rice_diseases_using_cnn_and_svm. 2019. Available online: https://www.kaggle.com/rajeshbhattacharjee/rice-diseases-using-cnn-and-svm (accessed on 24 February 2023).
36. 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.
37. Ding, W.; Taylor, G. Automatic moth detection from trap images for pest management. *Comput. Electron. Agric.* **2016**, *123*, 17–28. [CrossRef]
38. Dumoulin, V.; Visin, F. A guide to convolution arithmetic for deep learning. *arXiv* **2016**, arXiv:1603.07285.
39. 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.
40. Ioffe, S.; Normalization, C.S.B. Accelerating deep network training by reducing internal covariate shift. *arXiv* **2014**, arXiv:1502.03167.
41. Wang, S.H.; Muhammad, K.; Hong, J.; Sangaiah, A.K.; Zhang, Y.D. Alcoholism Identification via Convolutional Neural Network Based on Parametric ReLU, Dropout, and Batch Normalization. *Neural Comput. Appl.* **2020**, *32*, 665–680. [CrossRef]
42. Azulay, A.; Weiss, Y. Why Do Deep Convolutional Networks Generalize so Poorly to Small Image Transformations? *arXiv* **2019**, arXiv:1805.12177.
43. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
44. Jiang, F.; Lu, Y.; Chen, Y.; Cai, D.; Li, G. Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Comput. Electron. Agric.* **2020**, *179*, 105824. [CrossRef]
45. Jadhav, S.B.; Udupi, V.R.; Patil, S.B. Identification of plant diseases using convolutional neural networks. *Int. J. Inf. Technol.* **2021**, *13*, 2461–2470. [CrossRef]
46. Sethy, P.K.; Barpanda, N.K.; Rath, A.K.; Behera, S.K. Nitrogen deficiency prediction of rice crop based on convolutional neural network. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 5703–5711. [CrossRef]
47. Hossain, S.M.M.; Tanjil, M.M.M.; Ali, M.A.B.; Islam, M.Z.; Islam, M.S.; Mobassirin, S.; Sarker, I.H.; Islam, S.M.R. Rice Leaf Diseases Recognition Using Convolutional Neural Networks. In Proceedings of the Advanced Data Mining and Applications, Foshan, China, 12–14 November 2020; Yang, X., Wang, C.D., Islam, M.S., Zhang, Z., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, pp. 299–314. [CrossRef]
48. Hu, J.; Jia, C.; Liu, H.; Yi, X.; Liu, Y. A survey on state estimation of complex dynamical networks. *Int. J. Syst. Sci.* **2021**, *52*, 3351–3367. [CrossRef]
49. Hu, J.; Zhang, H.; Liu, H.; Yu, X. A survey on sliding mode control for networked control systems. *Int. J. Syst. Sci.* **2021**, *52*, 1129–1147. [CrossRef]
50. Tan, H.; Shen, B.; Peng, K.; Liu, H. Robust recursive filtering for uncertain stochastic systems with amplify-and-forward relays. *Int. J. Syst. Sci.* **2020**, *51*, 1188–1199. [CrossRef]
51. Li, X.; Han, F.; Hou, N.; Dong, H.; Liu, H. Set-membership filtering for piecewise linear systems with censored measurements under Round-Robin protocol. *Int. J. Syst. Sci.* **2020**, *51*, 1578–1588. [CrossRef]
52. Li, Q.; Liang, J. Dissipativity of the stochastic Markovian switching CVNNs with randomly occurring uncertainties and general uncertain transition rates. *Int. J. Syst. Sci.* **2020**, *51*, 1102–1118. [CrossRef]

remote sensing

MDPI

Article

Self-Attention and Convolution Fusion Network for Land Cover Change Detection Over a New Data Set in Wenzhou, China

Yiqun Zhu [1], Guojian Jin [1], Tongfei Liu [2,*], Hanhong Zheng [2], Mingyang Zhang [2], Shuang Liang [3], Jieyi Liu [2] and Linqi Li [1]

[1] Wenzhou Institute of Geotichnical Investigation and Surveying Co., Ltd., Wenzhou 325002, China
[2] School of Electronic Engineering, Xidian University, Xi'an 710121, China
[3] Academy of Advanced Interdisciplinary Research, Xidian University, Xi'an 710071, China
* Correspondence: ltfei@stu.xidian.edu.cn

Abstract: With the process of increasing urbanization, there is great significance in obtaining urban change information by applying land cover change detection techniques. However, these existing methods still struggle to achieve convincing performances and are insufficient for practical applications. In this paper, we constructed a new data set, named Wenzhou data set, aiming to detect the land cover changes of Wenzhou City and thus update the urban expanding geographic data. Based on this data set, we provide a new self-attention and convolution fusion network (SCFNet) for the land cover change detection of the Wenzhou data set. The SCFNet is composed of three modules, including backbone (local–global pyramid feature extractor in SLGPNet), self-attention and convolution fusion module (SCFM), and residual refinement module (RRM). The SCFM combines the self-attention mechanism with convolutional layers to acquire a better feature representation. Furthermore, RRM exploits dilated convolutions with different dilation rates to refine more accurate and complete predictions over changed areas. In addition, to explore the performance of existing computational intelligence techniques in application scenarios, we selected six classical and advanced deep learning-based methods for systematic testing and comparison. The extensive experiments on the Wenzhou and Guangzhou data sets demonstrated that our SCFNet obviously outperforms other existing methods. On the Wenzhou data set, the precision, recall and F1-score of our SCFNet are all better than 85%.

Keywords: computational intelligence; land cover/land use; change detection; self-attention; remote sensing images

Citation: Zhu, Y.; Jin, G.; Liu, T.; Zheng, H.; Zhang, M.; Liang, S.; Liu, J.; Li, L. Self-Attention and Convolution Fusion Network for Land Cover Change Detection Over a New Data Set in Wenzhou, China. *Remote Sens.* **2022**, *14*, 5969. https://doi.org/10.3390/rs14235969

Academic Editor: Parth Sarathi Roy

Received: 12 October 2022
Accepted: 18 November 2022
Published: 25 November 2022

1. Introduction

With the development of economy and science and technology, China's urbanization process has achieved a continuously significant increase [1]. One of the main features of the continuous acceleration of urbanization is the rapid expansion of urban land types and scales caused by the increase in urban population [2]. Therefore, the timely and effective detection of urban land use/cover changes has potential value for practical applications, such as dynamic monitoring of geographic conditions [3], urban development planning [4], and urban expansion trend analysis [5,6]. In this context, change detection techniques based on multi-temporal remote sensing images were applied to obtain quantitative or qualitative information on land use and land cover changes [7–10].

In recent decades, many change detection techniques have made remarkable progress. In the early stage, change detection can be achieved in two steps, i.e., difference image generation and difference image segmentation. Common difference image generation methods include image difference [11,12], image ratio [13,14], and change vector analysis (CVA) [15–17]. Difference map segmentation can usually be achieved by choosing a suitable threshold (e.g., Otsu [18]), or by using clustering algorithm (e.g., k-means [19,20], fuzzy

c-means [21], support vector machine (SVM) [22]). Accordingly, many methods have been widely used in practical applications [23]. For example, a method based on spectral CVA is applied to extract the change information of Wuhan city [24]. In [25], change detection and geographic information system based on remote sensing is used to analyze the land use changes during fifteen-year time period of 1991–2006; The change detection based on CVA is employed to acquire change information in Himachal Pradesh, India [26]; In [5], the author promoted a modified ratio operator to generate a change image; Urban change information can be obtained by using this method based on multitemporal synthetic aperture radar (SAR) images in Beijing and Shanghai, China; A land cover change detection method based on SVM was developed to map urban growth in the Algerian capital [27]. Various applications can be found in [28,29]. Although these approaches have been used in practical applications, they still require manual re-editing due to their low accuracy and efficiency. Moreover, with the popularization of very-high resolution (VHR) remote sensing images and rapid urban expansion, there is an urgent need to propose more timely and effective change detection methods to obtain more accurate information on land use and land cover changes [8,14].

With the popularity of deep learning (DL) technology in the field of computer vision, the technology has attracted continuous attention in the field of remote sensing [30–32]. Many DL-based methods have been applied to many remote sensing tasks, such as: change detection [33,34], hyperspectral classification [35,36], remote sensing scene classification [37], semantic segmentation [38], and object detection [39], etc. Under this situation, DL-based change detection has made some progress [40,41]. In the early stage, DL was used to achieve difference image segmentation in change detection due to its excellent classification performance. Zhao et al. proposed a deep neural network to classify the difference image into a binary change map [42]. Lei et al. promoted a change detection network for landslide inventory mapping [43]. The method was first to generate a difference image, and it was denoised by multivariate morphological reconstruction. Then, a fully convolutional network within pyramid pooling was devised to segment the difference image into a change map. In the following years, in order to avoid the noise introduced by traditional difference image generation methods, many DL-based methods are further proposed for change detection. For example, Gong et al. presented a novel DL-based change detection method, which can omit the process of a difference image generation. This method can effectively avoid using the traditional difference image generation method and reduce its adverse effect on the change map. Similarly, Lv et al. employed a dual-path fully convolutional network to directly obtain the landslide map without calculating the change magnitude image. The landslide mapping performance of this method was verified on real landslide sites on Lantau Island in Hong Kong, China. Although these DL-based methods have achieved significantly better performance than traditional methods, these methods are still limited by the amount of experimental data in the data set and are difficult to extend to various practical applications on a large scale.

In recent years, more advanced DL-based end-to-end change detection methods have been proposed to alleviate the limitation of the amount of data [40]. These methods usually implement end-to-end change detection by treating the change detection task as a semantic segmentation task. In [44], three architectures based on a fully convolutional network are presented for end-to-end change detection, including fully convolutional early fusion (FC-EF), fully convolutional Siamese concatenation (FC-Siam-Conc), and fully convolutional Siamese difference (FC-Siam-Diff). According to this, many researchers have proposed many advanced end-to-end change detection networks based on these architectures. In recent years, to further expand the application of DL-based change detection, many researchers have constructed and open-sourced many advanced change detection networks and the large data sets of many different application scenarios. For instance, Ji et al. opened a data set, named the WHU data set, which includes a high-quality multi-source data set for building extraction, building instance segmentation and building change detection [45]. Meanwhile, the paper proposed a Siamese U-Net (SiUnet)

for building extraction [45]. The network can also provide competitive results on the WHU data set. Chen et al. released a large-scale data set, named LEVIR-CD [46], which is composed of 637 Google Earth remote sensing image pairs of 1024 × 1024 (0.5 m/pixel). In [46], a Siamese spatial-temporal attention neural network is also devised and applied to the LEVIR-CD for building change detection. Similar large-scale data sets are S2looking in [47]. After that, many new models were proposed for these data sets. An attention-guided change detection network is devised for these data sets in [48], and devoted to achieve a better accuracy of building change detection. Liu et al. designed a Siamese local–global pyramid network (SLGPNet) and transfer learning for building change detection, which achieves excellent performance in detecting building changes [49]. The above studies have shown that deep learning-based change detection methods have made some progress in urban scenarios, especially building change detection. However, only developing a building change detection approach cannot satisfy the change detection requirements of urban land use and land cover in complex urban scenarios.

Recently, to further promote the practical application of DL-based change detection methods [50,51], some general urban change detection data sets containing changes in different ground objects were created and released. In [52], a Google Earth data set was published, which is a more challenging data set as it covers various changes in different cities in China (Beijing, Shenzhen, Chongqing, Wuhan, and Xi'an). Moreover, the paper also provided a deeply supervised image fusion network for this Google Earth data set and obtained a better detection performance. In addition, Peng et al. created a publicly VHR Google Earth data set (named Guangzhou data set), which covers the suburban areas of Guangzhou City [53]. For the Guangzhou data set, the changes are mainly caused by the urbanization process in China in the past decade, mainly including the following changes: buildings, waters, roads, farmland, bare land, forests, ships, etc. As the above large-scale urban change detection data set becomes available, more state-of-the-art (SOTA) methods have been proposed for the change detection task of complex urban scenes. For instance, a high-frequency attention Siamese network was proposed in [54], which can improve the performance by exploiting a high-frequency attention block; In [55], Fang et al., designed an SNUNet, which combines the Siamese network and the NestedUNet. The SNUNet can perform better than other SOTA change detection methods on a large-scale change detection data set with season-varying. In addition, transformer-based networks have reached SOTA performance in computer vision. Recently, transformer-based networks have attracted the attention of many researchers in the field of remote sensing, especially change detection. In this context, some transformer-based change detection networks have been proposed. A bitemporal image transformer (BIT) was developed for change detection [56], which can capture the contextual information within the spatial-temporal domain. This network can accomplish the SOTA performance compared to several recent attention-based models. Similar methods can be found in [57,58].

Despite the fact that these methods achieved convincing performance in many public urban change detection data sets, they currently face some limitations. Firstly, almost all of these SOTA approaches rely on a large number of labelled samples for network training. Secondly, in general, the performance of each method on different data sets is still not sufficiently stable. Finally, there is a lack of reliability validation for using these methods in practical applications. In this situation, two key points need to be noticed in the practical application of change detection [59].

- The usability and generalization of DL-based change detection methods in practical application scenarios still need to be verified.
- It is potentially meaningful to flexibly and comprehensively use one or more of the existing methods to meet the goal of real-change detection application scenarios.

In this paper, we create a new and challenging urban change detection data set oriented by practical applications, named the Wenzhou data set. The purpose of the Wenzhou data set is to achieve geographic surveying and mapping dynamic update by urban change detection, thereby providing a solid geographic information basis for the

development of Wenzhou's "smart city". Driven by this purpose, we systematically test and compare the existing popular SOTA approaches using the Wenzhou data set, including two classical methods (FC-EF [44] and FC-Siam-Conc [44]) and four SOTA methods (SiUnet [45], SNUNet [55], SLGPNet [49], and BIT [56]). In addition, in order to meet the performance requirements of the Wenzhou data set in practical applications, we propose a self-attention and convolution fusion network (SCFNet) by combining multiple existing change detection networks or modules. The SCFNet consists of three modules. First, the backbone network of our SCFNet is the local–global pyramid feature extractor in SLGPNet [49], which can effectively capture multi-scale features. Then, a self-attention and convolution fusion module (SCFM) [60] is employed to replace the position attention module in the backbone network. The SCFM aims to capture the non-local features. Finally, a residual refinement module (RRM) [61] is deployed after the output of our backbone network. The RRM is composed of multiple residual convolutions with different dilation rates, which can refine the initial change results at the original image scale. The significant contributions of this paper are summarized as follows:

(1) We created a new and challenging Wenzhou change detection data set, which is mainly used to acquire timely and effective land cover changes induced by urbanization in Wenzhou city, China. Based on the Wenzhou data set, we systematically tested the adaptability and performance of some existing popular and SOTA change detection approaches.
(2) We constructed a self-attention and convolution fusion network (SCFNet) for land cover change detection, which can integrate multiple existing change detection networks or modules to enhance the performance of the model further. The constructed SCFNet can basically meet the practical application requirements of land cover change detection in Wenzhou city, China.
(3) Compared with other SOTA methods, experiments on our created Wenzhou data set demonstrated that our SCFNet can acquire better and more balanced precision and recall. That is, the precision and recall both reach an accuracy of more than 85%. Furthermore, the effectiveness of our SCFNet is also validated on the public Guangzhou data set and achieves a good performance.

The rest of this paper is arranged as follows. In Section 2, the materials and methodology are described in detail. Section 3 presents the experiments and results. Finally, the conclusions and future works are provided in Section 5.

2. Materials and Methodology

In this section, we present a detailed presentation of the materials and methodology used in this study. First of all, the details of the study area and data set are described in Section 2.1. Subsequently, in Section 2.2, the methodology is introduced in detail. In particular, an overview of the constructed SCFNet is provided in Section 2.2.1. Sections 2.2.2 and 2.2.3 illustrate the SCFM and the RRM, respectively.

2.1. Study Area

In this paper, we chose Wenzhou city as the study area, as shown in Figure 1. Wenzhou city is located in the middle of the coastline of the Pacific Rim (approximately 18,000 km) in mainland China, in the southeast of Zhejiang Province. The urban area of Wenzhou is approximately 1054 square kilometers, with mountains, forests, water bodies, and various surface types. In recent years, with the rapid and stable development of Wenzhou's urbanization process, the urban landscape of Wenzhou city has undergone tremendous changes. Consequently, the research and application of the DL-based land cover change detection approach is performed to provide a geographic information basis for Wenzhou's "smart city" construction, natural resource management, and urban geographic dynamic update.

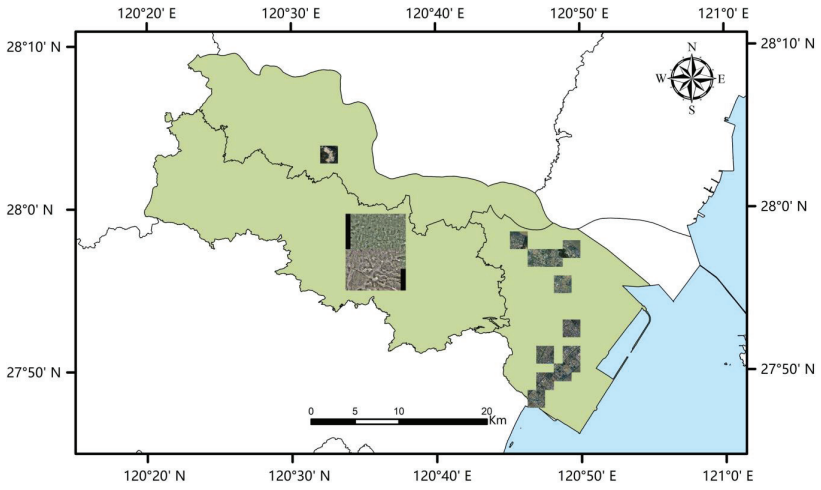


Figure 1. The spatial location of the study area of Wenzhou City, China.

In this study, we selected some representative areas (as shown in the rectangular area in Figure 1) from Wenzhou City to create our data set, named Wenzhou data set. Some representative examples of this data set are presented in Figure 2. The Wenzhou data set was captured between 2017 and 2021 by an aviation aircraft equipped with a Digital Mapping Camera III at an altitude of approximately 4.44 km. The spatial resolution was 0.2 m/pixel after re-sampling. This data set covers an area of approximately 112.026 square kilometers. The purpose of our created Wenzhou data set was to update the geographical data of urban expansion. Hence, it is mainly focused on land cover from natural objects to become related to urban construction areas (such as the changes in natural objects into buildings, bridges, roads, and other places related to urban expansion, without paying attention to changes in waters etc.). It is worth mentioning that the core changing features are built-up areas because of urbanization. The main challenges and requirements of this data set lie in the four following aspects.

(1) Bi-temporal images of the Wenzhou data set were collected from multiple periods (from 2017 to 2021). This may increase the difficulty of change detection since the bi-temporal images are shot under different atmospheric conditions, such as the sun height and moisture, etc.
(2) The changes in the built-up area of the Wenzhou data set are complex. Due to a large number of demolition and reconstruction projects in the Wenzhou urban area, the old and new houses in the old urban area and "urban villages" alternate, and high-rise buildings and low-rise buildings coexist. These conditions make land cover change detection in the Wenzhou data set more challenging.
(3) Since the primary type of change in the Wenzhou data set is a built-up area, and other types of changes are relatively small, this may lead to an imbalance in the number of different types of ground objects.
(4) To avoid secondary manual editing in practical applications, DL-based change detection methods require both precision and recall to be higher than 85%.

To sum up, according to the above characteristics, the Wenzhou data set is very suitable for systematically testing existing DL-based change detection methods. Furthermore, there is potential value in providing a reliable and satisfying solution for the Wenzhou data set. Hence, this study will further promote the practical application of DL-based change detection methods.

Figure 2. Some representative examples of the Wenzhou data set. (**a1**,**a2**) T_1-time image, (**b1**,**b2**) T_2-time image, and (**c1**,**c2**) ground truth image. White: changed pixels; Black: unchanged pixels.

2.2. Methodology

In this section, the proposed method is demonstrated in detail in three different parts. In the first part, the overall framework of SCFNet is briefly illustrated. In the second part, a mixed module of self-attention and convolution, SCFM, is introduced in detail. Finally, an employed performance refinement module, RRM, is illustrated in the third part.

2.2.1. Overview of Self-Attention and Convolution Fusion Network

A proper backbone is significant for correctly detecting building changes in the remote sensing data that are not perfectly orthophotos. Through extensive experiments, we found it difficult for many conventional state-of-the-art deep neural networks to acquire acceptable results over the new constructed data set. To tackle non-orthophoto bi-temporal images and the corresponding annotations, we employed a modified Siamese local–global pyramid network (SLGPNet) [49], which has been tested in similar tasks, as the backbone of the proposed SCFNet. The SLGPNet utilizes two different feature pyramids to better capture the local and global relationships between building objects over bi-temporal images, resulting in excellent results. Based on this fact, the encoder and decoder of SLGPNet are exploited in our work to acquire more accurate annotations of changed buildings over the study area. Additionally, another two network modules, SCFM and RRM, are introduced in the proposed network for finer performance.

Given the information below, the proposed method can be explained as follows: As shown in Figure 3, the bi-temporal remote sensing images are firstly concatenated and input into the local–global pyramid encoder to acquire the deep representative change information. Then, we exploit SCFM to refine the extracted feature through the fusion of the self-attention mechanism and convolutional layers. At the decoding stage, deep change information is gathered and integrated layer-by-layer. Finally, the change map is acquired after being refined by RRM.

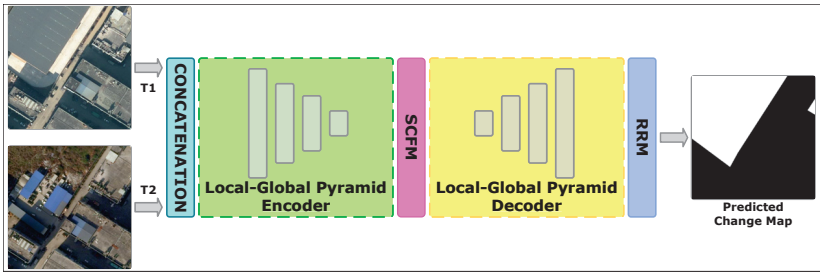


Figure 3. A brief illustration for proposed SCFNet. The SCFM and RRM indicate the self-attention convolution fusion module (SCFM) and residual refinement module (RRM), respectively.

2.2.2. Self-Attention and Convolution Fusion Module

The fusion of self-attention and convolutional layers have been proven helpful for deep learning-based image processing [60]. Inspired and encouraged by its success, similar techniques are introduced in the proposed method for better feature representation. In the SCFNet, the SCFM is employed to replace a self-attention-based module in the SLGPNet to better capture the semantic and location mapping of varied buildings in the study area, since there is an extra convolution path in the SCFM compared to the replaced module. Additionally, the SCFM can contribute to overcoming a specific challenge of the proposed data set, which is the commonly occurring non-orthophoto data. That is because there is a learnable shift operation-based convolution path in SCFM, which has the potential to better fit the non-orthophoto data set through the feature-level shift. As a result, the SCFM is introduced for a better feature representation and a finer annotation of non-orthophoto change information, and its brief process is depicted in Figure 4. With the illustration in Figure 4, the SCFM can be better described in the mathematical style below.

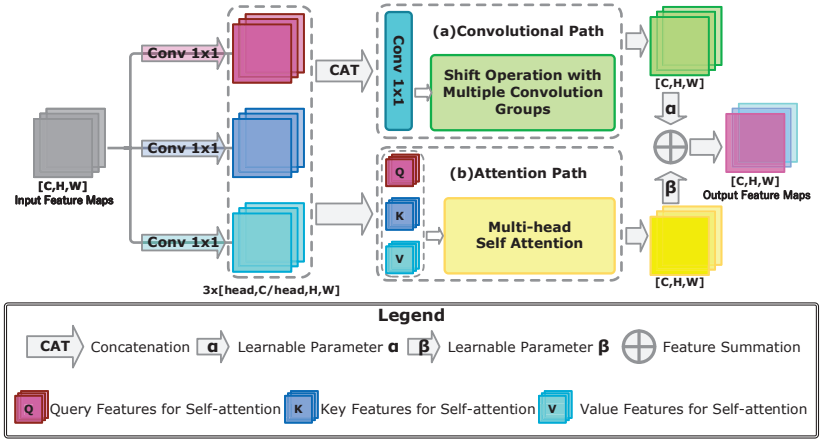


Figure 4. A brief illustration of the employed SCFM.

Firstly, the input feature maps of SCFM, $F_{input} \in \mathbb{R}^{C_{input} \times H \times W}$, comes from and was processed by the previous encoder layers of SCFNet, where $H \times W$, and C_{input} are the spatial and channel sizes of F_{input}, respectively. Then, F_{input} are transformed into three different parts with the size of $\mathbb{R}^{head \times C_{output}/head \times H \times W}$, which can be described as follows:

$$F_Q = Reshape\left(conv^1_{1\times1}(F_{input})\right) \tag{1}$$

$$F_K = Reshape\left(conv^2_{1\times1}(F_{input})\right) \tag{2}$$

$$F_V = Reshape\Big(conv^3_{1\times1}(F_{input})\Big) \tag{3}$$

where $\{conv^i_{1\times1}|i=1,2,3\}$ and Reshape indicates the convolutions with the kernel size of 1×1 and a shape transformation from $C_{output}\times H\times W$ to $head\times C_{output}/head\times H\times W$, respectively. The head represents the head number of multi-head attention in the SCFM, which is a fixed number of 4 in our method. At the next stage of SCFM, these features will be processed by two different paths, i.e., (a) convolutional path and (b) attention path, which can be illustrated as follows:

(a) Convolutional Path: In this path, features will be firstly gathered and projected by a feature concatenation and a 1×1 convolution, respectively. Then, a learnable shift operation will be conducted to the extracted feature maps, which is a multi-group convolutional layer with a set of reinitialized kernels. In this case, the extracted feature maps will firstly be shifted to several different fixed directions for a wider but rough cognition of non-orthophoto building objects. Then, the shift operation can be adjusted to a finer condition with these learnable kernels during supervised learning. The output of the convolutional path, $F_{conv}\in\mathbb{R}^{C_{output}\times H\times W}$, can be represented as follows:

$$F_{conv} = shift\_operation\Big(conv^4_{1\times1}\big(CAT(F_Q,F_K,F_V)\big)\Big) \tag{4}$$

where CAT indicates the feature concatenation, and $conv^4_{1\times1}$ represents a 1×1 convolutional layer. The $shift\_operation$ denotes the multi-group convolutional layer with the kernel size of 3.

(b) Attention Path: In the attention path, the extracted query, key, and value features are processed by a multi-head self-attention mechanism for a better feature representation, which can be briefly denoted as follows:

$$F_{att} = self\_attention(F_Q,F_K,F_V) \tag{5}$$

in which $F_{att}\in\mathbb{R}^{C_{output}\times H\times W}$ is the output of attention path in SCFM, and $self\_attention$ indicates the aforementioned multi-head self-attention with the head number of 4. Notably, positional encoding is also utilized in this stage for better location mapping.

With the output of both paths acquired, two learnable parameters are employed to generate $F_o\in\mathbb{R}^{C_{output}\times H\times W}$, and the final output of SCFM can be represented as:

$$F_o = \alpha * F_{conv} + \beta * F_{att} \tag{6}$$

where α and β are the learnable adjustment parameter for convolutional and attention paths, respectively. They are utilized to acquire a more stable and reliable output for SCFM.

2.2.3. Residual Refinement Module

In the proposed data set, large-scale building change areas are almost everywhere, which can be discovered in Figure 2. However, the predicted annotations can be incomplete for the deep learning-based method. More than that, in the application scene of this work, the completeness and correctness of the detected change areas are equally significant. Driven by this additional requirement, the RRM, which is inspired by [61], is introduced in the proposed method for more complete land cover detection. As shown in Figure 5, the RRM employs a series of dilated convolutions to refine the raw output of SCFNet to seek more complete annotations, which can be represented as outlined below.

Let $F_0\in\mathbb{R}^{H\times W}$ be the raw output waiting for the refinement of RRM, where H,W denotes the height and width, respectively. Then, a set of extracted features, $\{F_i\in\mathbb{R}^{32\times H\times W}\}$ where $\{i=1,2,3,4,5\}$, can be denoted as:

$$F_{i+1} = dilated\_conv^i_{3\times3}(F_i) \tag{7}$$

where $dilated\_conv^{i}_{3\times3}$ indicates 3×3 convolutions with different dilation rates. Then, these features are gathered and fused by a feature-wise summation and a convolutional layer, which can be demonstrated as:

$$F_m = conv_{3\times3}(F_1 + F_2 + F_3 + F_4 + F_5) \tag{8}$$

Finally, the refined output F_{ro} can be acquired as:

$$F_{ro} = F_m + F_0 \tag{9}$$

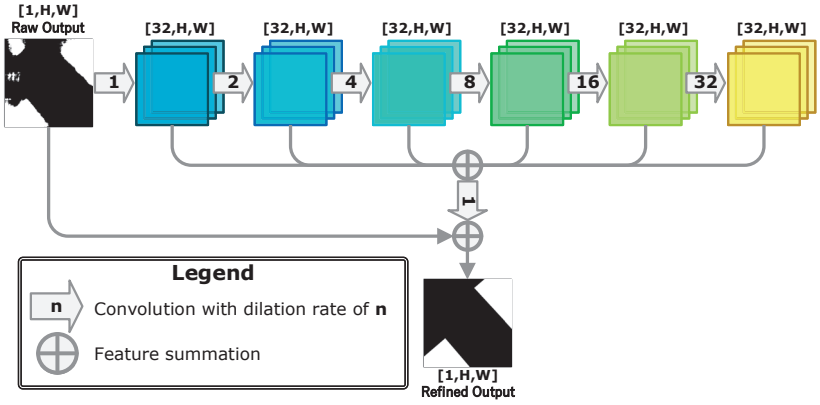


Figure 5. The structure of the RRM.

3. Experiments and Results

3.1. Experimental Settings

3.1.1. Data Set Descriptions

Wenzhou Data Set: For our created Wenzhou data set, to adapt the memory of the graphics card, the images for the entire study area are cropped into 4442 non-overlapping pairs of 512×512 pixels. We randomly divided all images into a training set (3554 tiles), a validation set (117 tiles), and a testing set (771 tiles). As such, all models were systematically tested and evaluated on the Wenzhou data set.

Guangzhou Data Set: This data set focuses on the land cover changes that occurred in the suburban areas of Guangzhou City, China, which share some similarities with the application scene in Wenzhou. Both of them depict the urbanization process that happened around the urban area. The remote sensing data of the Guangzhou data set is captured by Google Earth, between 2006 and 2019, with a spatial resolution of 0.55 m. In detail, it has 19 VHR bi-temporal image pairs with the sizes ranging from 1006×1168 to 4936×5224, which includes a large number of complicated scenes in different areas around Guangzhou. In our experiments, they are cropped into 3130 non-overlapping image pairs with the size of 256×256. We used 2191 of them for training. Furthermore, the rest of them are utilized as the testing data.

3.1.2. Evaluation Metrics

In the experiments, four widely used evaluation metrics were selected for the quantitative assessment and comparison of land cover change detection, including *Precision*, *Recall*, $F1 - Score$, and intersection over union (IoU) [49,54,56]. These four evaluation metrics can be calculated by the following formula.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

$$IoU = \frac{TP}{TP + FP + FN} \tag{13}$$

where TP, TN, FP, and FN denote true positive pixels, true negative pixels, false positive pixels, and false negative pixels, respectively. The confusion matrix can obtain TP, TN, FP, and FN based on the binary classification. Here, the *Precision* represents the proportion of correctly detected changed pixels among the detected as changed pixels. The *Recall* represents the proportion of correctly detected changed pixels among the truly changed pixels. The $F1 - score$ is an indicator that takes into account both precision and recall, because $F1$ can be regarded as the harmonic average of precision and recall. Additionally, the IoU represents the ratio of the intersection and union between pixels detected as changed and true changed pixels.

3.1.3. Benchmark Methods

To systematically evaluate and compare the performance of the existing DL-based change detection methods and our SCFNet, six benchmark methods were selected in the experiments. These approaches are presented as follows:

(1) FC-EF [44]: This method is a benchmark change detection model, which is a simplified U-shaped network. It employs an early fusion strategy to fuse bi-temporal images for change detection. This is a widespread end-to-end change detection framework.
(2) FC-Siam-Conc [44]: The model is also a U-shaped network. The difference is that it adopts a post-fusion strategy to fuse the features of bi-temporal images. Specifically, this model first extracts the deep features of the bi-temporal images by means of a Siamese encoder. Then, these deep features can be fused by the concatenation operation, and input into the decoder to obtain the change detection results. This is another attractive Siamese-based end-to-end change detection framework.
(3) SiUnet [45]: The method is a Siamese U-Net framework for building extraction. It uses a down-sampled counterpart of original bi-temporal images to enhance the multi-scale features of the network, resulting in improved detection performance. To this end, we adopted an early fusion strategy to deploy the SiUnet for the change detection task.
(4) SNUNet [55]: The model is constructed by the combination of Siamese network and NestedUNet, which can reduce the loss of localization information [55]. This method can achieve the SOTA performance on the CDD data set [55,62].
(5) SLGPNet [49]: This approach is an end-to-end Siamese-based building change detection network, which devises a local–global pyramid structure for building feature extraction. It obtains the best accuracy on WHU [45] and LEVIR-CD [46] data sets for change detection.
(6) BIT [56]: The model is a SOTA transformer-based change detection network. It exploits a transformer encoder and decoder to build the contexts within the spatial-temporal domain for change detection. This network acquires a promising performance on the LEVIR-CD [46], WHU [45], and DSIFN [52] data sets.

3.1.4. Implementation Details

In the experiments, all models were deployed based on the PyTorch platform. These models were trained on an NVIDIA RTX 3090 graphics card. The hyper-parameters of these benchmark methods are set to the optimal configuration. For our SCFNet, we employed the Adam optimizer with a weight decay rate of 1×10^{-5}, and the learning rate is initialized to 1×10^{-4}. Furthermore, binary cross entropy was adopted as the loss function for network training. The batch size of all models was set to 4 on both the Wenzhou and Guangzhou

data sets. It is worth noting that not all models exploit a data augmentation strategy. All models are trained and tested based on these settings for land cover change detection.

3.2. Results

3.2.1. Results on Wenzhou Data Set

As shown in Figure 6a, the confusion matrix of the proposed method is acquired on the Wenzhou data set. This confusion matrix indicates the overall performance of our method, especially on the changed and unchanged classes. Concretely, the quantitative results over Wenzhou data set indicate that the proposed method achieves an overwhelming advantage in all evaluation metrics compared to other benchmark methods, as listed in Table 1. Especially in IoU, the proposed SCFNet achieves the best performance of 75.36%, which is over 10% more than the second-best method. Moreover, both the Recall and Precision of SCFNet are over 85%, which achieves the requirement of this application scene in Wenzhou. Since our approach achieves the best recall and precision, it also has the best F1 performance over these benchmark methods, which suggests that our method can compete with current SOTA methods. These advantages in the Wenzhou data set can also be discovered in the corresponding visual results, as depicted in Figure 7. Generally, the proposed method can obtain more accurate change maps with less missed and false alarms. For example, in the fourth pair, the proposed SCFNet almost entirely detects two build-up areas with less false positive pixels than other methods. In this scene, BIT achieves a relatively low false alarm level, but the missed alarm is hard to ignore. To conclude, the proposed method outperforms these SOTA benchmark methods with significant advantages.

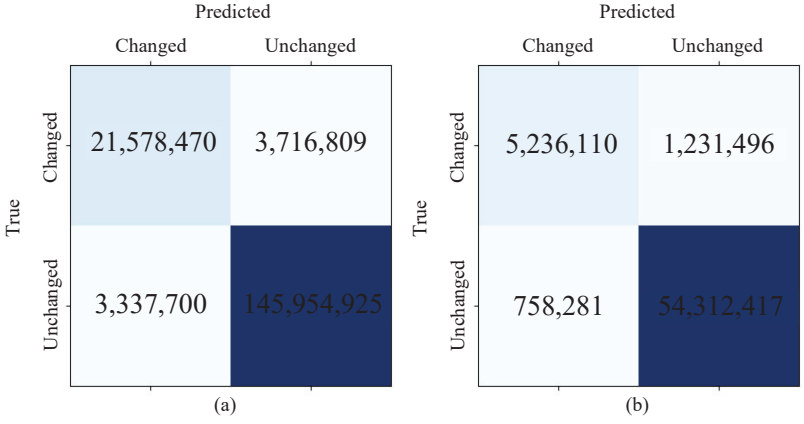


Figure 6. The confusion matrices of the results of the proposed SCFNet on two data sets. (**a**) Wenzhou data set; and (**b**) Guangzhou data set.

Table 1. Quantitative comparison of different methods on the Wenzhou data set.

Methods	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)
FC-EF [44]	67.14	56.24	61.21	44.10
FC-Siam-Conc [44]	52.39	53.18	52.79	35.85
SiUnet [45]	84.49	73.58	78.66	64.83
SNUNet [55]	73.83	61.33	67.00	50.38
SLGPNet [49]	78.39	75.84	77.09	62.72
BIT [56]	80.83	75.27	77.95	63.87
Proposed SCFNet	**86.60**	**85.31**	**85.95**	**75.36**

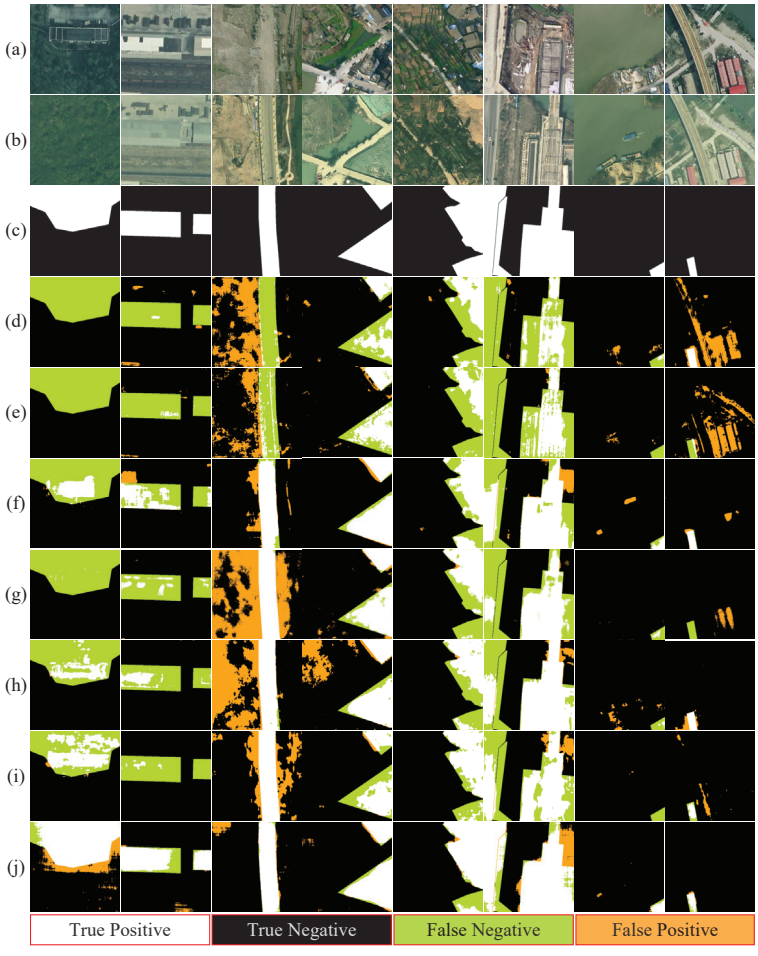


Figure 7. The results of the different methods on the Wenzhou data set: (**a**) T_1-time image; (**b**) T_2-time image; (**c**) ground truth image; (**d**) FC-EF [44]; (**e**) FC-Siam-Conc [44]; (**f**) SiUnet [45]; (**g**) SNUNet [55]; (**h**) SLGPNet [49]; (**i**) BIT [56]; and (**j**) proposed SCFNet.

3.2.2. Results on Guangzhou Data Set

As shown in Figure 6b, the confusion matrix of the proposed SCFNet is obtained on the Guangzhou data set, which shows the overall accuracy. In addition, the quantitative experimental results on the Guangzhou data set are listed in Table 2. In the aspects of main evaluation metrics, i.e., F1 and IoU, the proposed SCFNet still has significant advantages compared to other benchmark methods, which are over 1%. In terms of precision and recall, the performance advantages of SCFNet are not that significant. However, the proposed SCFNet can have both higher precision and recall, which can be challenging for other methods, thus contributing to the best F1 of SCFNet. In contrast, although BIT achieves the highest precision, it fails to achieve a higher F1 and IoU, since BIT has a relatively low recall performance. Similar conclusions can be discovered from the visual results shown in Figure 8. For instance, the proposed method can obtain more complete and accurate building annotations in the sixth pair of visual results over the Guangzhou data set. Generally, these visual results indicate that RRM helps the proposed method achieve a more complete annotation of changed land cover.

Table 2. Quantitative comparison of different methods on the Guangzhou data set.

Methods	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)
FC-EF [44]	77.62	56.97	65.71	48.94
FC-Siam-Conc [44]	83.02	55.42	66.47	49.78
SiUnet [45]	85.54	73.48	79.05	65.36
SNUNet [55]	49.17	50.00	49.58	32.96
SLGPNet [49]	85.25	80.88	83.00	70.95
BIT [56]	**87.86**	71.84	79.05	65.36
Proposed SCFNet	87.35	**80.96**	**84.03**	**72.46**

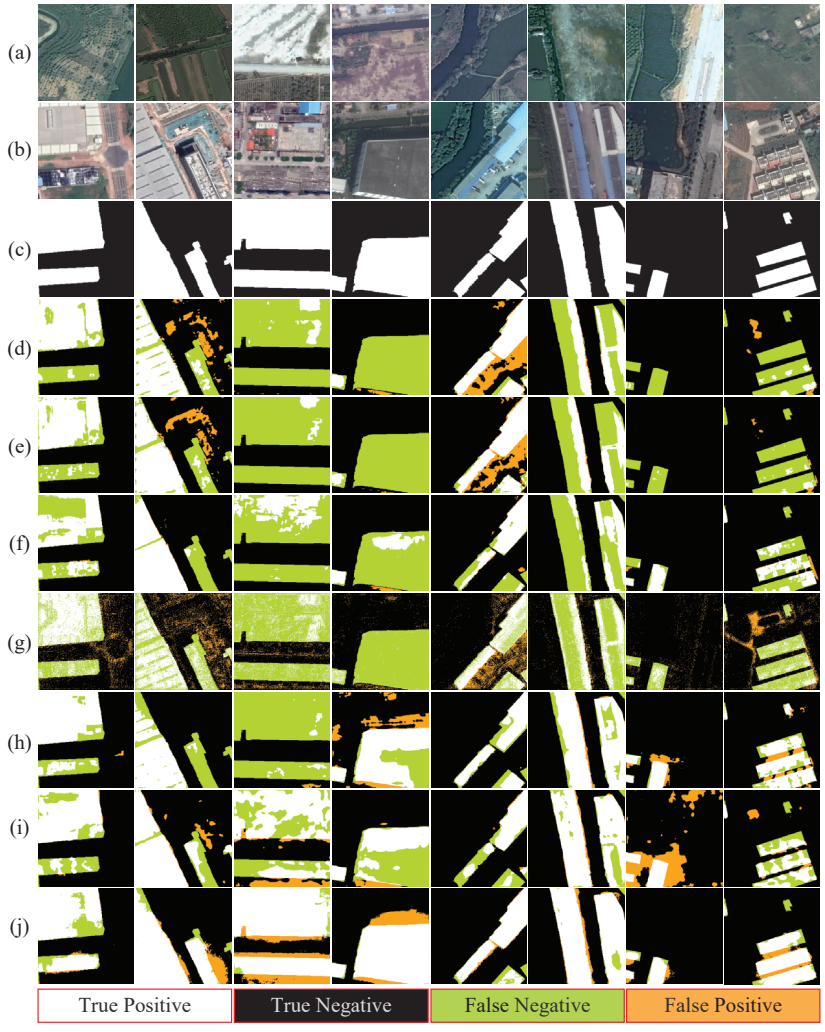


Figure 8. The results of different methods on Guangzhou data set: (**a**) T_1-time image; (**b**) T_2-time image; (**c**) ground truth image; (**d**) FC-EF [44]; (**e**) FC-Siam-Conc [44]; (**f**) SiUnet [45]; (**g**) SNUNet [55]; (**h**) SLGPNet [49]; (**i**) BIT [56]; and (**j**) proposed SCFNet.

3.3. Ablation Study

In our SCFNet, three modules, including backbone in SLGPNet [49], SCFM, and RRM, are integrated into the SCFNet for land cover change detection on the Wenzhou data set. Previous experimental results show that our SCFNet can achieve a convincing performance. In this section, we further implemented the ablation experiment on Wenzhou and Guangzhou data sets to analyze each component's effect in the SCFNet.

To achieve this, the quantitative results of networks with different module combinations were obtained for both data sets, as listed in Tables 3 and 4. For the experimental results in the Wenzhou data set, the accuracy obtained with the backbone alone is obviously insufficient. When the SCFM and the backbone were combined, the four evaluation indicators (precision, recall, F1-score, and IoU) were improved by 0.50%, 0.53%, 0.53%, and 0.74%, respectively. Here, SCFM only replaced the position attention module in the backbone, so the improvement obtained is slight. Similarly, the performance of combining the RRM and the backbone is more prominent. For example, compared with using backbone alone, the F1-score and IoU metrics were improved by 2.13% and 3.07%, respectively; compared with the network combining the backbone and the SCFM, the F1-Score and IoU metrics obtained 1.60% and 2.33% improvements, respectively. This is because the RRM can employ a larger receptive field to refine the initial change detection maps. According to this, the introduction of RRM can significantly improve the accuracy. Finally, when these three modules were deployed simultaneously, our SCFNet could achieve the best performance on four evaluation metrics. Notably, precision, recall and F1-score are higher than 85% after the full SCFNet is implemented for the Wenzhou data set.

Table 3. Quantitative evaluation of the combination of different modules on the Wenzhou data set.

Backbone	SCFM	RRM	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)
✓			85.79	76.80	81.04	68.13
✓	✓		86.29	77.33	81.57	68.87
✓		✓	85.04	81.39	83.17	71.20
✓	✓	✓	86.60	85.31	85.95	75.36

Table 4. Quantitative evaluation of the combination of different modules on the Guangzhou data set.

Backbone	SCFM	RRM	Precision (%)	Recall (%)	F1-Score (%)	IoU (%)
✓			85.27	79.40	82.23	69.82
✓	✓		84.35	82.02	83.17	71.19
✓		✓	83.54	83.91	83.72	72.01
✓	✓	✓	87.35	80.96	84.03	72.46

The experimental results on the Guangzhou data set report similar conclusions. The SCFM and RRM successfully improved the F1-score by 0.94% and 1.49% for the bare backbone in this data set, respectively. When used together, the complete SCFNet achieves the best F1-score in the Guangzhou data set. In addition, for a more intuitive comparison, Figure 9 presents the performance of different model combinations on different evaluation metrics. Figure 9 shows that our SCFNet combined with SCFM and RRM can effectively improve recall without reducing precision in the Wenzhou data set. To sum up, our SCFNet consists of these two modules in the existing network that can further improve the change detection performance.

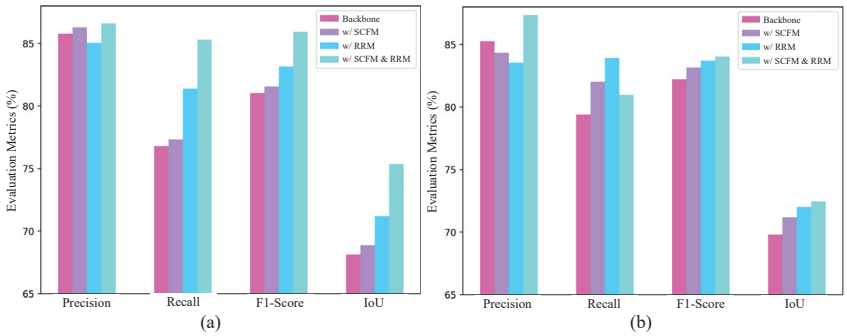


Figure 9. The performance comparison of the combination of different modules on different evaluation indicators for two data sets. (**a**) Wenzhou data set and (**b**) Guangzhou data set.

4. Discussion

To further discover the relation between the computational cost and performance for recent DL-based methods, we count the FLOPs and parameters (Params) of each model in Table 5. Basically, a model with higher computational costs usually leads to better performance. Although the proposed method has a higher computational cost, it achieves the best performance. Moreover, our SCFNet outperforms SLGPNet with a lower computational cost. Based on the computational cost and related performance shown in Table 5, we systematically discuss the performance of each benchmark method as follows:

(1) FC-EF [44] and FC-Siam-Conc [44]: FC-EF [44] can achieve a better performance than FC-Siam-Conc on the Wenzhou data set, while FC-Siam-Conc [44] has higher accuracy than FC-EF [44] on the Guangzhou data set. Overall, these two models performed poorly on both the Wenzhou and Guangzhou data sets. This is because the capacity of these two models is too small to handle complex data sets.

(2) SiUnet [45]: it achieves the second- and third-best performance on Wenzhou and Guangzhou data sets, respectively. The SiUnet [45] exploits the down-sampled counterpart of the original bi-temporal images as a branch of the Siamese network, enhancing the network's ability to represent multi-scale features. Hence, SiUnet [45] is a simple and effective model for the Wenzhou and Guangzhou data sets compared with other benchmark methods. This strategy is worthy of follow-up research.

(3) SNUNet [55]: Surprisingly, SNUNet [55] did not perform satisfactorily on the both Wenzhou and Guangzhou data sets. Although SNUNet [55] combines the Siamese network and NestedUNet to reduce the loss of localization, NestedUNet may introduce too many shallow features leading to incorrect semantic discrimination for facing the complex scene.

(4) SLGPNet [49]: SLGPNet [49] can reach a relatively stable accuracy on both the Wenzhou and Guangzhou data sets. This model is composed of a local–global pyramid feature extractor and a change detection head. The local–global pyramid feature extractor combines the position attention module, local feature pyramid, and global spatial pyramid, which has a robust multi-scale feature representation ability for change detection. However, the accuracy of this method still has some limitations for practical applications. The reason may be that the change detection head of this method contains only a few parameters, which makes the feature fusion of the final bi-temporal image insufficient for change detection.

(5) BIT [56]: Furthermore, BIT [56] is a SOTA transformer-based network for change detection. This model acquires the third-best and second-best accuracy on the Wenzhou and Guangzhou data sets, respectively. That is because BIT [56] can employ a transformer encoder to build the context of semantic tokens and exploit a Siamese transformer decoder to project semantic tokens into the pixel space for effective feature extraction.

Nonetheless, BIT [56] is difficult to balance between P and R. This limits the overall performance of BIT [56].

(6) Proposed SCFNet: Unlike the above methods, our SCFNet achieves the best performance on the both Wenzhou and Guangzhou data sets. Moreover, our SCFNet obtains precision and recall balanced accuracy on the Wenzhou data set, and its precision, recall, and F1-Score are higher than 85%. The core reasons include two aspects. First, the introduction of SCFM can improve the feature extraction capability of complex scenes. Second, the RRM deployed in SCFNet is able to refine the initial change results to obtain more accurate and complete change detection maps. Based on the above discussion, there are still some limitations in extending the existing methods to practical applications, such as the Wenzhou data set.

Table 5. Quantitative comparison of the performance (in F1-Score) and computational costs of different models.

Models	FLOPs (G)	Params (M)	Wenzhou (%)	Guangzhou (%)
FC-EF [44]	76.68	21.55	61.21	65.71
FC-Siam-Conc [44]	73.23	24.68	52.79	66.47
SiUnet [45]	185.08	31.05	78.66	79.05
SNUNet [55]	162.60	12.03	67.00	49.58
SLGPNet [49]	226.49	70.99	77.09	83.00
BIT [56]	17.54	3.50	77.95	79.05
Proposed SCFNet	212.23	72.85	85.95	84.03

According to the performance of our method, the comprehensive utilization of existing methods is an effective solution to promote DL-based change detection toward practical application. We hope this discussion provides a meaningful reference for subsequent related methods and applications.

5. Conclusions

This paper conducted an application-oriented study over the expanding built-up areas of Wenzhou City, China. A large scale of high-resolution bi-temporal remote sensing data was captured and annotated to obtain the land cover change information of Wenzhou between 2017 and 2021. With the help of these data, a new deep learning-based approach, SCFNet, was proposed for automatic land cover change detection over the study area. It employs the local–global pyramid encoder and decoder to build the backbone, and another two modules, i.e., SCFM and RRM, to further improve the performance. The SCFM combines the self-attention mechanism with convolutional layers to acquire a better feature representation. Furthermore, RRM employs dilated convolutions with different dilation rates to obtain more complete predictions over changed areas. In addition, a widely used open change detection data set, Guangzhou data set, and several current SOTA change detection methods were utilized to test the proposed method further. Furthermore, extensive experimental results indicated that SCFNet can outperform other benchmark methods in both large-scale data sets, i.e., the Wenzhou and Guangzhou data sets. As for future work, self-supervised and semi-supervised learning techniques can be utilized in our method to reduce the dependence on large-scale annotated data, which can lower the cost of collecting and constructing data.

Author Contributions: Conceptualization, Y.Z. and G.J.; methodology, Y.Z., G.J., T.L. and H.Z.; validation, T.L., H.Z. and M.Z.; investigation, Y.Z., G.J. and M.Z.; writing—original draft preparation, T.L. and H.Z.; writing—review and editing, Y.Z., G.J., J.L., S.L. and L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Shaanxi Province in China under Grant No. 2021JQ-209 and No. 2020JQ-313; the Fundamental Research Funds for the Central Universities, Grant No. JB210210 and No. XJS210216.

Data Availability Statement: Not applicable.

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

References

1. Yu, Z.; Yao, Y.; Yang, G.; Wang, X.; Vejre, H. Spatiotemporal patterns and characteristics of remotely sensed region heat islands during the rapid urbanization (1995–2015) of Southern China. *Sci. Total. Environ.* **2019**, *674*, 242–254. [CrossRef] [PubMed]
2. Liu, F.; Zhang, X.; Murayama, Y.; Morimoto, T. Impacts of land cover/use on the urban thermal environment: A comparative study of 10 megacities in China. *Remote Sens.* **2020**, *12*, 307. [CrossRef]
3. Ridd, M.K.; Liu, J. A comparison of four algorithms for change detection in an urban environment. *Remote Sens. Environ.* **1998**, *63*, 95–100. [CrossRef]
4. Wang, N.; Li, W.; Tao, R.; Du, Q. Graph-based block-level urban change detection using Sentinel-2 time series. *Remote Sens. Environ.* **2022**, *274*, 112993. [CrossRef]
5. Ban, Y.; Yousif, O.A. Multitemporal spaceborne SAR data for urban change detection in China. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2012**, *5*, 1087–1094. [CrossRef]
6. Lv, Z.; Wang, F.; Cui, G.; Benediktsson, J.A.; Lei, T.; Sun, W. Spatial-Spectral Attention Network Guided With Change Magnitude Image for Land Cover Change Detection Using Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–12. [CrossRef]
7. Sun, Y.; Lei, L.; Guan, D.; Li, M.; Kuang, G. Sparse-constrained adaptive structure consistency-based unsupervised image regression for heterogeneous remote-sensing change detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–14. [CrossRef]
8. Lv, Z.; Liu, T.; Benediktsson, J.A.; Falco, N. Land cover change detection techniques: Very-high-resolution optical images: A review. *IEEE Geosci. Remote Sens. Mag.* **2021**, *10*, 44–63. [CrossRef]
9. Viana, C.M.; Girão, I.; Rocha, J. Long-term satellite image time-series for land use/land cover change detection using refined open source data in a rural region. *Remote Sens.* **2019**, *11*, 1104. [CrossRef]
10. Sun, Y.; Lei, L.; Li, X.; Sun, H.; Kuang, G. Nonlocal patch similarity based heterogeneous remote sensing change detection. *Pattern Recognit.* **2021**, *109*, 107598. [CrossRef]
11. Bruzzone, L.; Prieto, D.F. Automatic analysis of the difference image for unsupervised change detection. *IEEE Trans. Geosci. Remote Sens.* **2000**, *38*, 1171–1182. [CrossRef]
12. Lv, Z.; Liu, T.; Zhang, P.; Atli Benediktsson, J.; Chen, Y. Land cover change detection based on adaptive contextual information using bi-temporal remote sensing images. *Remote Sens.* **2018**, *10*, 901. [CrossRef]
13. Lu, D.; Mausel, P.; Brondizio, E.; Moran, E. Change detection techniques. *Int. J. Remote Sens.* **2004**, *25*, 2365–2401. [CrossRef]
14. Ban, Y.; Yousif, O. Change detection techniques: A review. *Multitemporal Remote Sens.* **2016**, 19–43.
15. Liu, S.; Du, Q.; Tong, X.; Samat, A.; Bruzzone, L.; Bovolo, F. Multiscale morphological compressed change vector analysis for unsupervised multiple change detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 4124–4137. [CrossRef]
16. Zhuang, H.; Deng, K.; Fan, H.; Yu, M. Strategies combining spectral angle mapper and change vector analysis to unsupervised change detection in multispectral images. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 681–685. [CrossRef]
17. ZhiYong, L.; Wang, F.; Xie, L.; Sun, W.; Falco, N.; Benediktsson, J.A.; You, Z. Diagnostic analysis on change vector analysis methods for LCCD using remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 10199–10212. [CrossRef]
18. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man. Cybern.* **1979**, *9*, 62–66. [CrossRef]
19. Celik, T. Unsupervised change detection in satellite images using principal component analysis and *k*-means clustering. *IEEE Geosci. Remote Sens. Lett.* **2009**, *6*, 772–776. [CrossRef]
20. Lv, Z.; Liu, T.; Shi, C.; Benediktsson, J.A.; Du, H. Novel land cover change detection method based on K-means clustering and adaptive majority voting using bitemporal remote sensing images. *IEEE Access* **2019**, *7*, 34425–34437. [CrossRef]
21. Shao, P.; Shi, W.; He, P.; Hao, M.; Zhang, X. Novel approach to unsupervised change detection based on a robust semi-supervised FCM clustering algorithm. *Remote Sens.* **2016**, *8*, 264. [CrossRef]
22. Bovolo, F.; Bruzzone, L.; Marconcini, M. A novel approach to unsupervised change detection based on a semisupervised SVM and a similarity measure. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 2070–2082. [CrossRef]
23. Lv, Z.; Wang, F.; Sun, W.; You, Z.; Falco, N.; Benediktsson, J.A. Landslide Inventory Mapping on VHR Images via Adaptive Region Shape Similarity. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [CrossRef]
24. Xiaolu, S.; Bo, C. Change detection using change vector analysis from Landsat TM images in Wuhan. *Procedia Environ. Sci.* **2011**, *11*, 238–244. [CrossRef]
25. Singh, P.; Khanduri, K. Land use and land cover change detection through remote sensing & GIS technology: Case study of Pathankot and Dhar Kalan Tehsils, Punjab. *Int. J. Geomat. Geosci.* **2011**, *1*, 839–846.
26. Singh, S.; Sood, V.; Taloor, A.K.; Prashar, S.; Kaur, R. Qualitative and quantitative analysis of topographically derived CVA algorithms using MODIS and Landsat-8 data over Western Himalayas, India. *Quat. Int.* **2021**, *575*, 85–95. [CrossRef]
27. Nemmour, H.; Chibani, Y. Multiple support vector machines for land cover change detection: An application for mapping urban extensions. *ISPRS J. Photogramm. Remote Sens.* **2006**, *61*, 125–133. [CrossRef]

28. Lv, Z.; Liu, T.; Shi, C.; Benediktsson, J.A. Local histogram-based analysis for detecting land cover change using VHR remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 1284–1287. [CrossRef]
29. Liu, T.; Gong, M.; Jiang, F.; Zhang, Y.; Li, H. Landslide Inventory Mapping Method Based on Adaptive Histogram-Mean Distance With Bitemporal VHR Aerial Images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
30. Wen, D.; Huang, X.; Bovolo, F.; Li, J.; Ke, X.; Zhang, A.; Benediktsson, J.A. Change detection from very-high-spatial-resolution optical remote sensing images: Methods, applications, and future directions. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 68–101. [CrossRef]
31. Shafique, A.; Cao, G.; Khan, Z.; Asad, M.; Aslam, M. Deep learning-based change detection in remote sensing images: A review. *Remote Sens.* **2022**, *14*, 871. [CrossRef]
32. 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]
33. Wu, Y.; Li, J.; Yuan, Y.; Qin, A.K.; Miao, Q.G.; Gong, M.G. Commonality Autoencoder: Learning Common Features for Change Detection From Heterogeneous Images. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 4257–4270. [CrossRef] [PubMed]
34. Gong, M.; Jiang, F.; Qin, A.K.; Liu, T.; Zhan, T.; Lu, D.; Zheng, H.; Zhang, M. A Spectral and Spatial Attention Network for Change Detection in Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–14. [CrossRef]
35. Lv, Z.; Li, G.; Jin, Z.; Benediktsson, J.A.; Foody, G.M. Iterative training sample expansion to increase and balance the accuracy of land classification from VHR imagery. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 139–150. [CrossRef]
36. Wu, Y.; Mu, G.; Qin, C.; Miao, Q.; Ma, W.; Zhang, X. Semi-supervised hyperspectral image classification via spatial-regulated self-training. *Remote Sens.* **2020**, *12*, 159. [CrossRef]
37. Gong, M.; Li, J.; Zhang, Y.; Wu, Y.; Zhang, M. Two-Path Aggregation Attention Network With Quad-Patch Data Augmentation for Few-Shot Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–16. [CrossRef]
38. Zhang, J.; Lin, S.; Ding, L.; Bruzzone, L. Multi-scale context aggregation for semantic segmentation of remote sensing images. *Remote Sens.* **2020**, *12*, 701. [CrossRef]
39. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
40. Jiang, H.; Peng, M.; Zhong, Y.; Xie, H.; Hao, Z.; Lin, J.; Ma, X.; Hu, X. A Survey on Deep Learning-Based Change Detection from High-Resolution Remote Sensing Images. *Remote Sens.* **2022**, *14*, 1552. [CrossRef]
41. Shi, W.; Zhang, M.; Zhang, R.; Chen, S.; Zhan, Z. Change detection based on artificial intelligence: State-of-the-art and challenges. *Remote Sens.* **2020**, *12*, 1688. [CrossRef]
42. Zhao, J.; Gong, M.; Liu, J.; Jiao, L. Deep learning to classify difference image for image change detection. In Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 411–417.
43. Lei, T.; Zhang, Y.; Lv, Z.; Li, S.; Liu, S.; Nandi, A.K. Landslide inventory mapping from bitemporal images using deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 982–986. [CrossRef]
44. Daudt, R.C.; Le Saux, B.; Boulch, A. Fully convolutional siamese networks for change detection. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 4063–4067.
45. 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.* **2018**, *57*, 574–586. [CrossRef]
46. Chen, H.; Shi, Z. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote Sens.* **2020**, *12*, 1662. [CrossRef]
47. Shen, L.; Lu, Y.; Chen, H.; Wei, H.; Xie, D.; Yue, J.; Chen, R.; Lv, S.; Jiang, B. S2Looking: A satellite side-looking dataset for building change detection. *Remote Sens.* **2021**, *13*, 5094. [CrossRef]
48. Song, K.; Jiang, J. AGCDetNet: An attention-guided network for building change detection in high-resolution remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4816–4831. [CrossRef]
49. Liu, T.; Gong, M.; Lu, D.; Zhang, Q.; Zheng, H.; Jiang, F.; Zhang, M. Building Change Detection for VHR Remote Sensing Images via Local–Global Pyramid Network and Cross-Task Transfer Learning Strategy. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–17. [CrossRef]
50. Zhan, Y.; Fu, K.; Yan, M.; Sun, X.; Wang, H.; Qiu, X. Change detection based on deep siamese convolutional network for optical aerial images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1845–1849. [CrossRef]
51. Yang, L.; Chen, Y.; Song, S.; Li, F.; Huang, G. Deep Siamese networks based change detection with remote sensing images. *Remote Sens.* **2021**, *13*, 3394. [CrossRef]
52. Zhang, C.; Yue, P.; Tapete, D.; Jiang, L.; Shangguan, B.; Huang, L.; Liu, G. A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2020**, *166*, 183–200. [CrossRef]
53. Peng, D.; Bruzzone, L.; Zhang, Y.; Guan, H.; Ding, H.; Huang, X. SemiCDNet: A semisupervised convolutional neural network for change detection in high resolution remote-sensing images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5891–5906. [CrossRef]
54. Zheng, H.; Gong, M.; Liu, T.; Jiang, F.; Zhan, T.; Lu, D.; Zhang, M. HFA-Net: High frequency attention siamese network for building change detection in VHR remote sensing images. *Pattern Recognit.* **2022**, *129*, 108717. [CrossRef]
55. Fang, S.; Li, K.; Shao, J.; Li, Z. SNUNet-CD: A Densely Connected Siamese Network for Change Detection of VHR Images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]

56. Chen, H.; Qi, Z.; Shi, Z. Remote sensing image change detection with transformers. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–14. [CrossRef]
57. Zheng, Z.; Zhong, Y.; Tian, S.; Ma, A.; Zhang, L. ChangeMask: Deep multi-task encoder-transformer-decoder architecture for semantic change detection. *ISPRS J. Photogramm. Remote Sens.* **2022**, *183*, 228–239. [CrossRef]
58. Zhang, C.; Wang, L.; Cheng, S.; Li, Y. SwinSUNet: Pure Transformer Network for Remote Sensing Image Change Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–13. [CrossRef]
59. Zhou, S.; Dong, Z.; Wang, G. Machine-Learning-Based Change Detection of Newly Constructed Areas from GF-2 Imagery in Nanjing, China. *Remote Sens.* **2022**, *14*, 2874. [CrossRef]
60. Pan, X.; Ge, C.; Lu, R.; Song, S.; Chen, G.; Huang, Z.; Huang, G. On the integration of self-attention and convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 19–22 June 2022; pp. 815–825.
61. Shao, Z.; Tang, P.; Wang, Z.; Saleem, N.; Yam, S.; Sommai, C. BRRNet: A fully convolutional neural network for automatic building extraction from high-resolution remote sensing images. *Remote Sens.* **2020**, *12*, 1050. [CrossRef]
62. Lebedev, M.; Vizilter, Y.V.; Vygolov, O.; Knyaz, V.; Rubis, A.Y. Change detection in remote sensing images using conditional adversarial networks. In Proceedings of the ISPRS TC II Mid-term Symposium "Towards Photogrammetry 2020", Riva del Garda, Italy, 4–7 June 2018; Volume 2.

remote sensing

MDPI

Article

Frequency Spectrum Intensity Attention Network for Building Detection from High-Resolution Imagery

Dan Feng *, Hongyun Chu and Ling Zheng

School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
* Correspondence: fengdan@xupt.edu.cn

Abstract: Computational intelligence techniques have been widely used for automatic building detection from high-resolution remote sensing imagery and especially the methods based on neural networks. However, existing methods do not pay attention to the value of high-frequency and low-frequency information in the frequency domain for feature extraction of buildings in remote sensing images. To overcome these limitations, this paper proposes a frequency spectrum intensity attention network (FSIANet) with an encoder–decoder structure for automatic building detection. The proposed FSIANet mainly involves two innovations. One, a novel and plug-and-play frequency spectrum intensity attention (FSIA) mechanism is devised to enhance feature representation by evaluating the informative abundance of the feature maps. The FSIA is deployed after each convolutional block in the proposed FSIANet. Two, an atrous frequency spectrum attention pyramid (AFSAP) is constructed by introducing FSIA in widely used atrous spatial pyramid pooling. The AFSAP is able to select the features with high response to building semantic features at each scale and weaken the features with low response, thus enhancing the feature representation of buildings. The proposed FSIANet is evaluated on two large public datasets (East Asia and Inria Aerial Image Dataset), which demonstrates that the proposed method can achieve the state-of-the-art performance in terms of F1-score and intersection-over-union.

Keywords: computational intelligence; building detection; attention mechanism; remote sensing image

Citation: Feng, D.; Chu, H.; Zheng, L. Frequency Spectrum Intensity Attention Network for Building Detection from High-Resolution Imagery. *Remote Sens.* **2022**, *14*, 5457. https://doi.org/10.3390/rs14215457

Academic Editor: Mohammad Awrangjeb

Received: 9 September 2022
Accepted: 28 October 2022
Published: 30 October 2022

1. Introduction

With the development of satellite, aviation, and unmanned aerial vehicle (UAV) technology, huge amounts of high-resolution (HR) remote sensing images have been captured in a constant stream [1–3]. These HR remote sensing images have been applied to land cover classification [4–6], change detection [7–9], target recognition [10,11], and image restoration and registration [12,13], for example. This brings opportunities for us to observe fine objects such as buildings, roads, vehicles, etc. Among them, buildings are one of the most important targets in the surface coverage of remote sensing images. Therefore, building detection or extraction has become a hot topic of study, as it plays a crucial role in digital city construction and management [11,14,15] and sustainable urban development [16,17], among other applications.

Although building detection has made some progress in recent years, the widespread use of HR remote sensing images from different sensors has brought new challenges to this task [18,19]. These challenges include mainly the following:

(a) A large number of fine ground targets can be depicted by very-high-resolution aerial imagery, e.g., trees, roads, vehicles, and swimming pools, etc. However, these targets often easily interfere with the identification of buildings due to their similar features (e.g., spectrum, shape, size, structure, etc.).

(b) In urban areas, tall buildings often have severe geometric distortions caused by fixed sensor imaging angles. This may lead to accurate building detection becoming challenging.
(c) With the rapid development of urbanization, many cities and rural areas are interspersed with tall buildings and short buildings. Tall buildings often exhibit large shadows when imaged by the sun. This phenomenon may not only make it difficult to accurately detect tall buildings themselves, but may also obscure other features (especially short buildings), thus limiting the effective detection of buildings.

Recently, deep-learning-based building detection techniques have been introduced to alleviate these challenges to some extent [20]. State-of-the-art (SOTA) methods are able to improve the performance of building detection through a variety of techniques, including the introduction of multi-scale modules [21,22], edge information [23,24], and attention mechanisms [25,26]. For instance, Ji et al. proposed a Siamese U-Net (SiU-Net) for building extraction, which can enhance multi-scale feature extraction by adding a branch with a small resolution downsampled input image [19]. In [27], a named Building Residual Refine Network (BRRNet) was designed to achieve accurate and complete building extraction. This network is composed of a prediction module and a residual refinement module. In the prediction module, an atrous convolution is employed to capture multi-scale global features. The residual refinement module can refine the initial result of the prediction module, thereby obtaining a more accurate and complete building detection. Yang et al. promoted an edge-aware network, which consists of image segmentation networks and edge perception networks [28]. The network combines the network with edge-aware loss to achieve better performance.

These previous networks have achieved good detection results. Some methods effectively enhance the feature characterization ability of the network by some attention or multi-scale operations, thus improving the detection effect. Some recent approaches propose the introduction of edge information (edge module or edge loss supervision) to help building recognition. However, there are still some limitations to overcome. First, supervised learning strategies by introducing edge loss directly outside the network structure can lead to difficult convergence and less stable results. Second, the combination of roughly applied edge information and convolutional networks is both difficult to be well embedded in the neural network and prone to introduce some interference information from other ground target edges. Finally, edge information tends to represent only high-frequency information of buildings, whereas low-frequency information is equally important in pixel-level prediction tasks. Therefore, enhancing both high-frequency and low-frequency information can further improve the building feature characterization ability.

To address the aforementioned issues, our solutions are motivated by the following two aspects. On the one hand, Zheng et al. proposed a high frequency attention Siamese network for building change detection [29]. The study has verified that the introduction of high frequency information can enhance the network's ability to sense buildings. However, introducing frequency domain information directly in the building detection task can easily introduce interference information from other features, thus limiting the building feature extraction. For this reason, inspired by this approach, we perform feature enhancement by introducing the attention module of the global feature map with frequency domain information. In particular, the average frequency spectral intensity of an image can express the amount of high frequency information contained in the image as a whole. This can effectively evaluate the features that are more conducive to building extraction. Therefore, the introduction of average frequency spectral intensity will be beneficial to building detection tasks. In this case, building detection performance may be further improved when both high-frequency and low-frequency information are considered in the network. On the other hand, atrous spatial pyramid pooling (ASPP) is often used to capture multi-scale features in remote sensing image understanding [30,31]. However, different building features can be obtained by using atrous convolution with different atrous rates. In this context, it would enhance the building feature representation if the features with high response to

the building semantic features at each scale are emphasized while the features with low response are weakened. According to these motivations, we propose a frequency spectrum intensity attention network (FSIANet) for building detection. The major contributions of this paper include the following three aspects:

(1) This paper proposes a novel computational intelligence approach for automatic building detection, named FSIANet. In the proposed FSIANet, we devised a plug-and-play FSIA without the requirement of learnable parameters. The FSIA mechanism based on frequency–domain information can effectively evaluate the informative abundance of the feature maps and enhance feature representation by emphasizing more informative feature maps. To this end, The FSIANet can significantly improve the building detection performance.

(2) An atrous frequency spectrum attention pyramid (AFSAP) is devised in the proposed FSIANet. It is able to mine multi-scale features. At the same time, by introducing FSIA in ASPP, it can emphasize the features with high response to building semantic features at each scale and weaken the features with low response, which will enhance the building feature representation.

(3) The experimental results on two large public datasets (Inria [18] and East Asia [19]) have demonstrated that the proposed FSIANet can achieve a more effective building detection compared to other classical and SOTA approaches.

The remainder of this article is arranged as follows. Section 2 reviews the relevant literature. Methodology and experiments are presented in Sections 3 and 4. Finally, Section 6 concludes this article.

2. Related Work

In the past decade, building detection and roof extraction has been a hot research topic in the field of remote sensing. In the early stage, some handcrafted building features are used to implement building detection and extraction, such as pixel shape index [32], morphological profiles [33], etc. For example, Huang et al. combined the information of the morphological building index and the morphological shadow index for building extraction. Other morphological building index-based methods are available in [34–36]; Bi et al. proposed a multi-scale filtering building index to reduce the noise of building map in [21]. Although relying on these early hand-made building features can extract buildings from HR impacts, these methods are still poor in terms of accuracy and completeness of building detection and extraction.

With the rapid development of deep learning technology, deep learning has been extensively extended to the field of remote sensing. So far, deep-learning-based building detection approaches have become the most advanced technology. In the early stage, researchers treated the building detection task as an image segmentation task. Therefore, semantic segmentation networks widely used in computer vision can be directly applied to achieve building detection tasks, such as fully convolutional network (FCN) [37], U-Net [38], SegNet [39], etc. The introduction of these deep-learning-based methods leads to a significant improvement in the performance of building detection and extraction compared to hand-crafted feature methods. Nonetheless, with the unprecedented increase in the spatial resolution of images, researchers still found some new challenges, that is, buildings with large or small scales are difficult to accurately identify due to the local receptive fields of convolutional neural networks (CNN).

To overcome the above limitation, many multi-scale CNN have further promoted computer vision [40]. For instance, Zhao et al. designed a pyramid scene parsing network (PSPNet) for semantic segmentation [41]. In the PSPNet [41], a pyramid pooling module is used to capture global features, thereby improving the multi-scale feature extraction capability of the network. In [42], an atrous spatial pyramid pooling (ASPP) is devised to effectively enlarge the receptive field of the network, thereby improving the multi-scale feature representation ability of the network. These multi-scale CNN in computer vision have also been developed in the field of remote sensing [43,44]. Wang et al. promoted a

novel FCN for dense semantic labeling [45]. This network can effectively mine multi-scale features by combining the advantages of both encoder-decoder and ASPP. Yu et al. applied an end-to-end segmentation network for pixel-level building detection, which combines the ASPP and skip connections generative adversarial segmentation network to aggregate multi-scale contextual information [31]. Similar research also includes [46–48].

In recent years, attention mechanisms have been widely used in deep learning [9,49–51], especially computer vision. Attention mechanisms commonly used in computer vision and remote sensing image processing can be divided into two major categories according to the function of the attention mechanism [52,53]: channel attention and spatial attention. Channel attention aims to enhance the feature representation ability of the network by selecting important feature channels [54–56]. Spatial attention is able to generate an attention mask in the spatial domain and employ it to emphasize the most task-relevant spatial regions [57,58]. In addition to multi-scale CNN, driven by the attention mechanism, it is another effective technique to improve the performance of building detection. For instance, spatial and channel attention mechanisms are simultaneously used to emphasize spatial regions and feature channels with high semantic responses to buildings, thereby improving the capability of the building feature extraction [59]. In [60], a pyramid attention network (PANet) is promoted to achieve pixel-level semantic segmentation; an encoder-decoder network based on attention-gate and ASPP (AGPNet) is proposed for building detection from UAV images [25]; Guo et al. [61] devised a scene-driven multi-task parallel attention network to overcome the large intraclass variance of buildings in different scenes; other attention-based methods are available in [62,63]. Recently, many experts have designed some novel networks dedicated to automatic building detection and extraction. Transformer-based methods are the latest and most compelling new network structures. Wang et al. promoted a vision transformer network for building extraction [44]. A transformer-based multi-scale feature learning network was proposed in [64]. In addition, a new deep architecture, named Res2-Unet, was proposed for building detection [65]. This architecture is an end-to-end structure, which can exploit multi-scale learning at a granular level to extend the receptive field. These methods further advance the development of building detection.

In summary, although some progress has been made in previous work, there are still certain limitations that need to be further addressed. In particular, there is a lack of research on the role of frequency–domain information in building detection tasks. For one thing, the combination of roughly applied edge information and convolutional networks is both difficult to be well embedded in the neural network and prone to introduce some interference information from other ground target edges. For another thing, edge information tends to represent only high-frequency information of buildings, whereas low-frequency information is equally important in pixel-level prediction tasks.

3. Methodology

In this section, the detailed information of the proposed method will be given. First, a brief overview of the proposed FSIANet and the overall procedure will be illustrated in Section 3.1. Second, Section 3.2 will explain the proposed frequency spectrum intensity attention (FSIA) mechanism in detail. Finally, the atrous frequency spectrum attention pyramid (AFSAP) will be demonstrated in Section 3.3.

3.1. Overview of FSIANet

In Figure 1, the framework and overall inference process are illustrated. As shown in the figure, the raw HR remote sensing data are first input into the input layer of FSIANet. Subsequently, the initially extracted feature maps will be input into the down-sample layers followed by FSIA. With the network going deeper, the size of feature maps will be smaller, which contain the semantic and location information of land cover depicted on the input HR images. Then the deepest features will be improved by the proposed AFSAP. At the next stage, the previously extracted feature maps will be gradually gathered and processed

by the up-sample layers with FSIA. Introducing previous features can significantly improve the performance of similar networks, which was demonstrated in [38]. During this stage, the spatial and semantic information of different levels will be integrated and fused to annotate building-like land cover at the output layer.

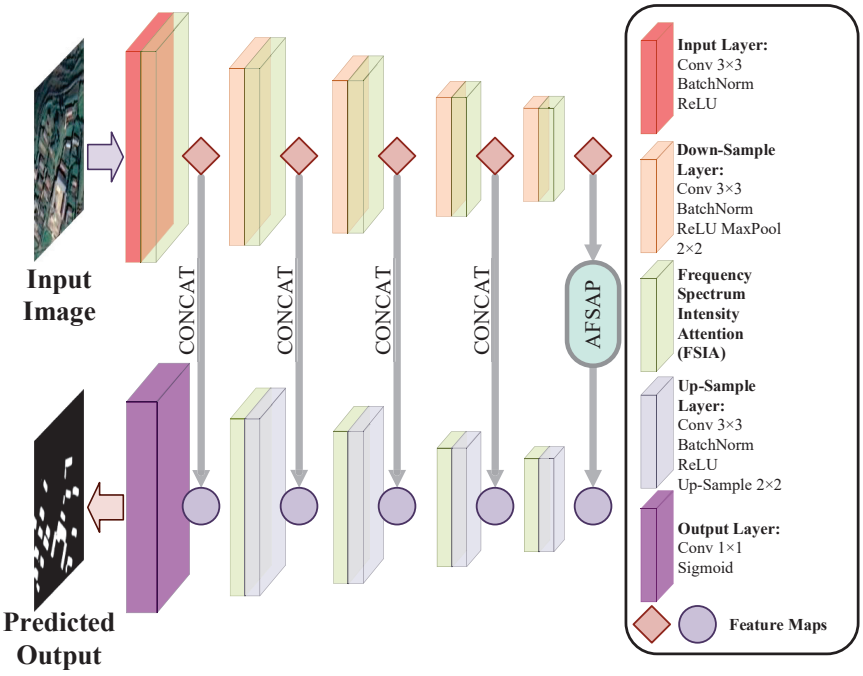


Figure 1. The brief procedure of the proposed FSIANet. The AFSAP indicates the proposed atrous frequency spectrum attention pyramid.

3.2. Frequency Spectrum Intensity Attention

Because attention mechanisms can bring potential performance improvement for deep-learning-based methods, they have been successfully utilized in many remote sensing tasks. However, most of the existing attention modules can reach a satisfying performance only after long-period training with networks. In addition, introducing frequency domain information, which can benefit the performance [29], is usually neglected in most network-based remote sensing methods. According to these facts, a new parameterless frequency-aware attention mechanism can be potential beneficial for deep-learning-based methods. To avoid these conventional problems, a novel attention mechanism, FSIA, is proposed for a better representation of building-like objects in our FSIANet. It aims for better feature representation without extra parameters waiting to be trained. As shown in Figure 2, the FSIA relies on frequency domain information to evaluate the importance of each extracted feature map and thereby enhance them accordingly. Based on the previous description, its mathematical representation can be demonstrated as follows:

First, let $F^I \in \mathbb{R}^{C\times H\times W}$ be the input features, in which C, H, and W represent the channel, height, and width sizes, respectively. The frequency spectrum of F^I, $F^S \in \mathbb{R}^{C\times H\times W}$, can be denoted as:

$$F^S = DCT\left(F^I\right) \tag{1}$$

where $DCT(\cdot)$ is the channel-wise discrete cosine transformation, which acquires the frequency domain information. Then the global frequency information vector $V^S \in \mathbb{R}^{C\times1\times1}$ can be obtained by:

$$V^S = GAP\left(F^S\right) \tag{2}$$

where $GAP(\cdot)$ denotes the global average pooling. The global frequency spectrum intensity of each channel can be quantified through this way. To significantly enhance the informative feature maps, a channel-wise Softmax function is applied as follows:

$$V^A = Softmax\left(V^S\right) \tag{3}$$

where $Softmax(\cdot)$ indicates the Softmax function, whereas $V^A \in \mathbb{R}^{C\times1\times1}$ represents the channel-wise attention score. Given the attention weight V^A, the final output of FSIA, $F^O \in \mathbb{R}^{C\times H\times W}$, can be given as:

$$F^O = F^I \bigotimes V^A \bigoplus F^I \tag{4}$$

in which $\bigotimes$ and $\bigoplus$ demonstrate a channel-wise multiplication and a pixel-wise addition, respectively. In conclusion, FSIA tries to achieve a better feature representation in a unique parameterless pipeline, which is introduced in the frequency information. It is exploited numerous times in the proposed method, as it can be applied to features of any spatial size.

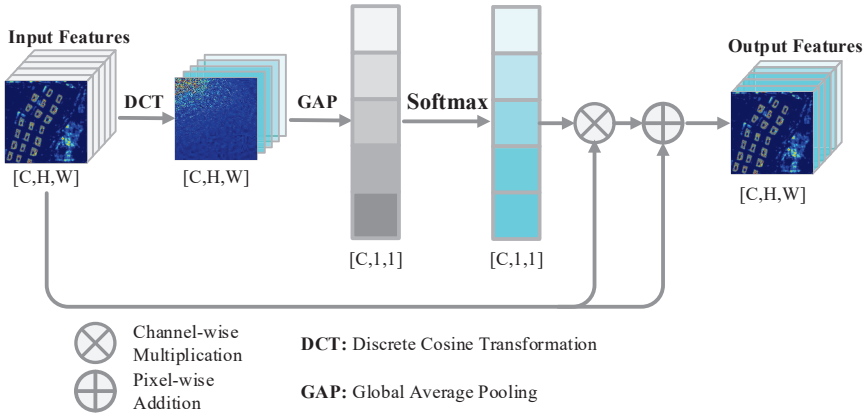


Figure 2. The procedure of FSIA.

3.3. Atrous Frequency Spectrum Attention Pyramid

Except for the accurate semantic recognition of buildings, acquiring precise geographical locations and scales is also significant for fine building annotation in HR images. According to existing related work, multi-scale feature pyramids can help deep-learning-based methods better recognize land cover objects of various scales. In our work, we also propose an attention-based feature pyramid, AFSAP, to obtain better building annotation when dealing with multi-scale objects. Inspired by ASPP, atrous convolution with different dilation rates and global average pooling are utilized in AFSAP to obtain the features with different reception fields. Based on these features, proposed FSIA is employed to acquire finer feature representation, which is able to acquire higher performance improvement compared to bare ASPP. The detailed demonstration of AFSAP is shown in Figure 3. Its detailed process can be represented as the following equations:

Let $F^D \in \mathbb{R}^{C\times H\times W}$ be the deepest features of FSIANet. Then the features with different reception fields $F_i^{RF} \in \mathbb{R}^{256\times H\times W}\{i = 1, 2, 3, 4, 5\}$ can be obtained as follows:

$$F_1^{RF} = Conv_{1\times 1}^1\left(F^D\right) \tag{5}$$

$$F_2^{RF} = AsConv_{3\times 3}^1\left(F^D\right) \tag{6}$$

$$F_3^{RF} = AsConv_{3\times 3}^2\left(F^D\right) \tag{7}$$

$$F_4^{RF} = AsConv_{3\times 3}^3\left(F^D\right) \tag{8}$$

$$F_5^{RF} = interpolation\left(Conv_{1\times 1}^2\left(GAP\left(F^D\right)\right)\right) \tag{9}$$

where $Conv_{1\times 1}^1(\cdot)$ and $Conv_{1\times 1}^2(\cdot)$ indicate the convolutional layers with the kernel size of 1×1, which are followed by batch normalization (BN) and ReLU function. In addition, $AsConv_{3\times 3}^1(\cdot)$, $AsConv_{3\times 3}^2(\cdot)$, and $AsConv_{3\times 3}^3(\cdot)$ represent 3×3 atrous convolution with dilation rates of 6, 12, and 18, respectively. These atrous convolutional layers are also followed by BN and ReLU. The expression $interpolation(\cdot)$ is the bilinear interpolation that reverts feature size to $H \times W$. At the next stage, these extracted features F_i^{RF} are distilled by FSIA and gathered in channel dimension as follows:

$$\dot{F}_i^{RF} = FSIA\left(F_i^{RF}\right) \tag{10}$$

$$\widetilde{F}^{RF} = Concat\left(\dot{F}_1^{RF}, \dot{F}_2^{RF}, \dot{F}_3^{RF}, \dot{F}_4^{RF}, \dot{F}_5^{RF}\right) \tag{11}$$

With $\widetilde{F}^{RF}$ acquired, the output of AFSAP can be represented as:

$$\widetilde{F}^D = Conv_{1\times 1}^3\left(\widetilde{F}^{RF}\right) \tag{12}$$

where $Conv_{1\times 1}^3(\cdot)$ is a convolutional layer with the kernel size of 1×1, which is used to integrate and refine the collected features.

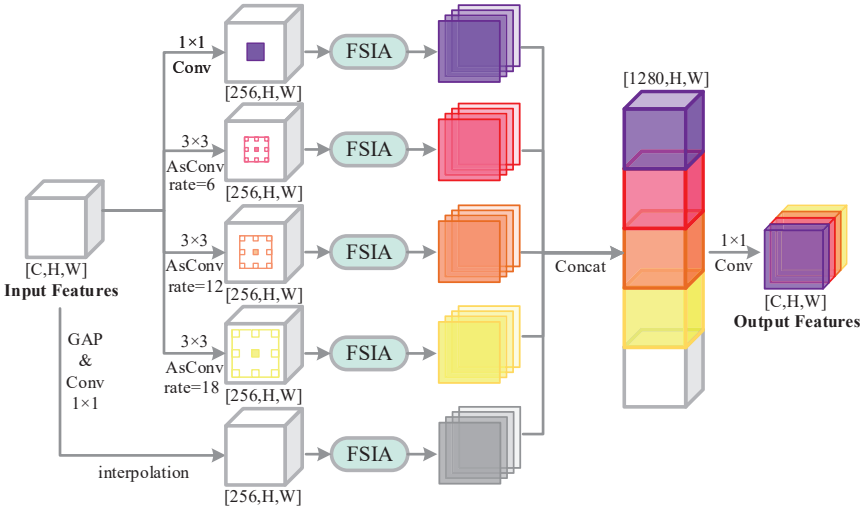


Figure 3. The procedure of AFSAP.

As a summary for AFSAP, the proposed feature pyramid can acquire better recognition for various buildings with the help of multi-scale reception fields provided by atrous convolutions. The proposed FSIA can facilitate and improve the feature extraction and representation of AFSAP, which gives AFSAP the ability to outperform ASPP.

4. Experimental Results and Analysis

In this section, we first briefly introduce three benchmark datasets and measurement indicators required for all experiments. The implementation details of the proposed FSIANet are also given. Subsequently, we will show the experimental results compared with other excellent peers. The ablation experiments of our proposed FSIANet are also analyzed in depth.

4.1. Dataset Descriptions and Evaluation Metrics

In this paper, two commonly used building detection datasets, East Asia Dataset [19] and Inria Aerial Image Dataset [18], are employed in the experiments to fairly validate the effectiveness of all methods. The detailed information of these datasets is presented in Table 1. Furthermore, some examples of these two datasets are shown in Figure 4. It is worth noting that we have processed both benchmark datasets accordingly on the basis of the original datasets.

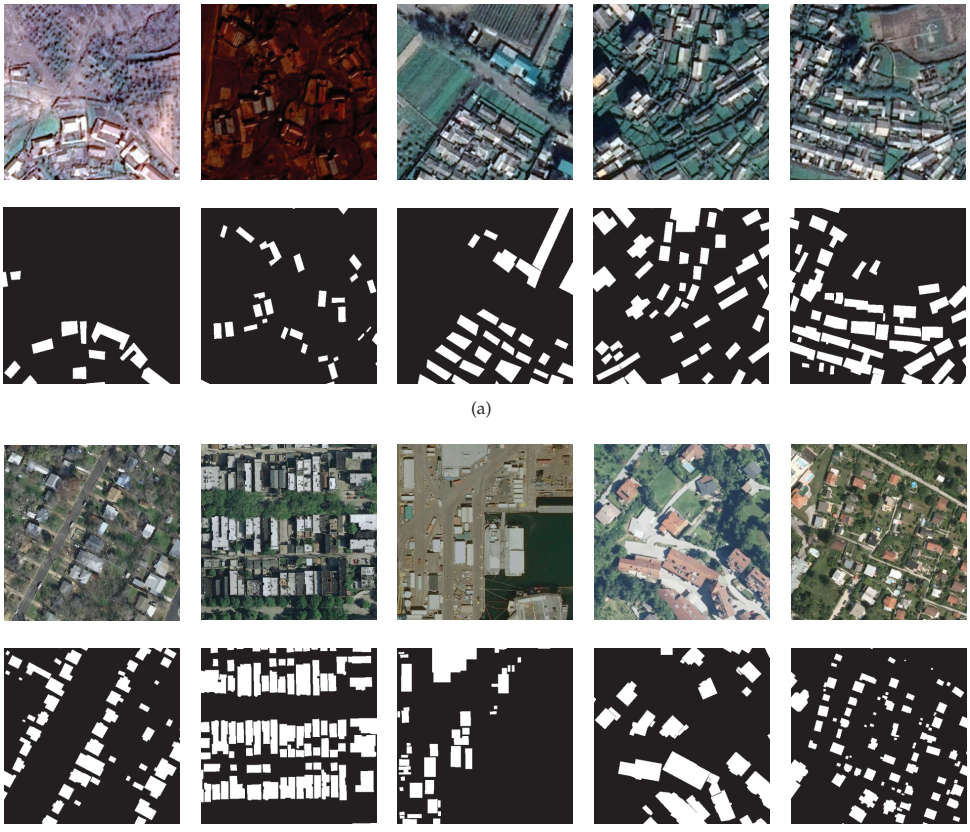


Figure 4. Some examples of two benchmark datasets. (**a**) East Asia Dataset. (**b**) Inria Aerial Image Dataset. The first row in each subplot is the aerial image tile, and the second row is the ground truth.

Table 1. The detailed information of the two building detection datasets.

Dataset	East Asia Dataset	Inria Aerial Image Dataset
Year	2019	2017
Coverage	550 km^2	810 km^2
Size	512 × 512 pixels	5000 × 5000 pixels
Spatial Resolution	2.7 m	0.3 m

East Asia Dataset [19] is a sub-dataset of the WHU Building Dataset, which consists of six neighboring satellite images in East Asia. The vector building map was completely hand-drawn in ArcGIS software and contained a total of 34,085 buildings. Specifically, 3153 and 903 aerial image tiles are selected as training and test sets, respectively. This East Asia Dataset is primarily used to evaluate and develop the generalization ability of deep learning models to different data sources but with similar architectural styles in the same geographic area. Therefore, this is recognized as one of the most challenging building extraction datasets.

We perform all the experiments with a total of 180 aerial image tiles covering an area of 405 km^2 for the Inria Aerial Image Dataset [18]. It contains a total of five sub-datasets, namely Austin, Chicago, Kitsap, Tyrol, and Vienna, each of which consists of 36 aerial image tiles. We take the first 25 aerial image tiles and the remaining 11 aerial image tiles in each sub-dataset as a training set and a testing set, respectively. Consistent with [19,66], we crop all the aerial images to a size of 512 × 512 pixels. Therefore, the training and test sets in each sub-dataset consist of 2025 and 891 aerial images, respectively. The Inria Aerial Image Dataset was collected at different times and places. It is a very challenging task to accurately extract buildings with huge differences in architectural style, structure, and distribution in each place.

In terms of evaluation metrics, four commonly used building extraction indicators, namely *Precision*, *Recall*, *F1-Score*, and *Intersection over Union*(IoU), are employed for pixel-based evaluation to measure the performance of all methods. By convention, TP and TN represent the number of true positive and true negative pixels, respectively; FP and FN denote the number of false positive and false negative pixels, respectively. Based on this, *Precision* refers to the percentage of area that is predicted to be correct for buildings, which is defined as follows:

$$Precision(P) = \frac{TP}{TP + FP}. \tag{13}$$

The value *Recall* represents the proportion of positive examples in the building ground truths that is predicted to be correct, which can be calculated as follows:

$$Recall(R) = \frac{TP}{TP + FN}. \tag{14}$$

The *F1-Score*, a comprehensive indicator, is the harmonic mean of precision and recall, so it can be obtained as follows:

$$F1\text{-}Score(F1) = \frac{2 \times R \times P}{R + P}. \tag{15}$$

The IoU, also a comprehensive evaluation indicator, represents the ratio of the intersection area over the union area between the ground truths and the building predictions, which can be obtained as follows:

$$IoU = \frac{TP}{TP + FN + FP}. \tag{16}$$

4.2. Implementation Details

In order to ensure the fairness of the comparison, we reproduce all peers and conduct all the experiments under the following execution conditions. It is worth noting that none of the deep learning models adopt strategies such as data augmentation or pre-training that can improve the performance of building extraction. This can ensure that the above interference is eliminated to the greatest extent, and the reason for the improvement is attributed to the proposed modules or strategies. Specifically, we implemented the experiments on a NVIDIA GTX 3090 based on the Pytorch framework in CUDA 11.6. In terms of parameter setting, we employed the Adam optimizer and the multistep learning rate delay, where the initial learning rate is set to 0.0001. In Adam, the coefficients used to calculate the moving average of the gradient and its square are set to 0.9 and 0.999, respectively. In addition, the batch size is set to 4.

4.3. Comparison with Other Methods

4.3.1. Comparative Algorithms

To demonstrate the effectiveness of our proposed method, seven outstanding peers are selected as comparative methods, and their detailed introductions are as follows:

(1) FCN8s [37] (2015): This work includes three classic convolutional neural network characteristics, i.e., a fully convolutional network that discards the fully connected layer to adapt to the input of any size image; deconvolution layers that increase the size of the data enable it to output refined results; and a skip-level structure that combines results from different depth layers while ensuring robustness and accuracy.
(2) U-Net [38] (2015): The proposed U-Net is an earlier model that applies convolutional neural networks to image semantic segmentation, which is built on the basis of FCN8s [37]. U-Net includes contracting paths to extract image features or context and expanding paths for accurate segmentation.
(3) PSPNet [41] (2017): PSPNet mainly extracts multi-scale information through pyramid pooling, which can better extract global context information and utilize both local and global information to make scene recognition more reliable.
(4) PANet [60] (2018): PANet proposed a pyramid attention network to exploit the influence of global contextual information in semantic segmentation, combining an attention mechanism and a spatial pyramid to extract precise pixel-annotated dense features instead of using complex diffuse convolution and hand-designed decoder networks.
(5) SiU-Net [19] (2019): The East Asia Dataset was released in [19]. In addition, SiU-Net is designed with a Siamese fully convolutional network, in which two branches of the network share weights, and the original image and its downsampled counterpart are taken as inputs.
(6) BRRNet [27] (2020): The prediction module and residual refinement module are the main innovations of BRRNet. The prediction module obtains a larger receptive field by introducing atrous convolutions with different dilation rates. The residual refinement module takes the output of the prediction module as input.
(7) AGPNet [25] (2021): This is a SOTA ResNet50-based network, which combines grid-based attention gate and ASPP for building detection. This method is similar to ours and is valuable for comparing methods.
(8) Res2-Unet [65] (2022): Res2-Unet employed granular-level multi-scale learning to expand the receptive field size of each bottleneck layer, focusing on pixels in the border region of complex backgrounds.

4.3.2. Results on the East Asia Dataset

Table 2 shows the quantitative experimental results of *Precision*, *Recall*, *F*1-*Score*, and *IoU* on the East Asia Dataset. Similar to the results on the Inria Aerial Image Dataset, FSIANet does not perform as well as other comparison algorithms on *Precision*, but achieves the best results on *Recall*. In fact, the two are contradictory in some cases. For ex-

ample, in the extreme case where there are only a very small number of buildings, we only predict one result and it is accurate, then the *Precision* is 100%, but the *Recall* is very low, and vice versa. Therefore, two composite indicators, *F*1-*Score* and *IoU*, should be given priority consideration. It can be concluded from Table 2 that FSIANet outperforms the SOTA algorithm (i.e., BRRNet) by 1.88% and 2.69% on *F*1-*Score* and *IoU*, respectively. Similarly, compared with AGPNet [25], the proposed FSIANet achieves 1.2% and 1.72% improvement on F1 and IoU. The improvement of FSIANet on building detection is mainly attributed to the FSIA mechanism based on frequency domain information, which can effectively evaluate the information abundance of feature maps and enhance feature representation by emphasizing more informative feature maps.

Table 2. Quantitative results on *Precision*, *Recall*, *F*1-*Score*, and *IoU* (in %) of different methods on the East Asia Dataset. The best results are shown in bold.

Methods	***Precision***	***Recall***	***F1-Score***	***IoU***
FCN8s [37]	87.30	70.32	77.90	63.79
U-Net [38]	88.41	71.22	78.89	65.14
PSPNet [41]	83.66	69.97	76.20	61.56
PANet [60]	87.69	64.09	74.05	58.80
SiU-Net [19]	**89.09**	69.76	78.25	64.27
BRRNet [27]	83.06	78.11	80.51	67.37
AGPNet [25]	86.37	76.59	81.19	68.34
Res2-Unet [65]	84.07	69.14	75.88	61.14
FSIANet (Ours)	84.11	**80.75**	**82.39**	**70.06**

We also provide some visualization results in the East Asia Dataset to further illustrate the effectiveness of our proposed FSIANet. The related visualization comparisons are shown in Figure 5. In the case shown in Figure 5, the buildings in the yellow boxes are not obvious, and there are trees, shadows, and other disturbances around. Algorithms such as FCN8s and PANet have difficulty extracting the approximate building outlines. This is largely because they focus too much on local information and are sensitive to parameters, and their attention mechanisms lack the connection between global information. Res2-Unet, PSPNet, and BRRNet also have certain missed detections. Compared with other methods, the buildings extracted by FSIANet are more accurate and clear on the whole.

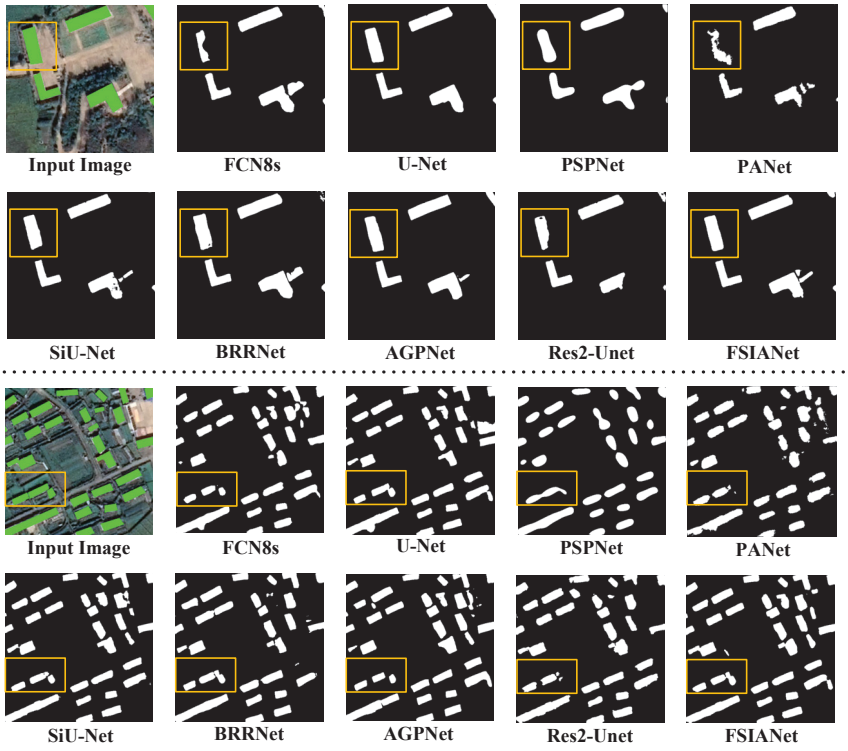


Figure 5. The visualization results of the proposed FSIANet and other comparison methods on the East Asia Dataset.

4.3.3. Results on the Inria Aerial Image Dataset

The experimental results of the four indicators on the Inria Aerial Image Dataset are shown in Table 3. In terms of *Precision*, the proposed FSIANet has less obvious advantages compared with other algorithms. However, due to the extreme imbalance of positive and negative samples in many aerial images in the Inria Aerial Image Dataset, the proportion of buildings in some scenes is very low. Therefore, higher accuracy does not mean that the performance of the algorithm for extracting buildings is better. As such, the excellent performance of FSIANet on *Recall* is also not convincing. Based on this, we have to focus on the performance of the methods on two comprehensive indicators, i.e., *F*1-*Score* and *IoU*. On these two metrics, FSIANet achieves the best experimental results, with an overall improvement of 0.45% in *F*1-*Score* and 0.71% in *IoU* compared to the existing SOTA methods. Specifically, the improvement of FSIANet is most obvious in the Kitsap and Tyrol regions. It is worth noting that there is a huge gap in the distribution of aerial image buildings in these two regions, with both dense and sparse building scenes. It can be explained that the proposed FSIANet has strong generalization performance to apply in various complex scenarios.

In addition to the experimental results of the quantitative analysis, we also present some representative visualizations of the Inria Aerial Image Dataset. Figure 6 shows the results of binary prediction visualizations of our FSIANet and seven other comparison methods in the Austin, Chicago, Kitsap, Tyrol, and Vienna regions. As in the aerial image example shown in Figure 6, the Inria dataset has some images with very low proportions of buildings. For illustration purposes, we mark the more visible regions with yellow rectangles. It can be concluded from Figure 6 that our proposed FSIANet method outperforms other methods overall, especially in recognizing edge, tiny, and

shadow buildings. Furthermore, we can conclude from the examples of moderately dense buildings in Austin and Vienna that FSIANet performs well in the connection of multiple complex buildings. This is because the porous spectral attention pyramid is capable of mining multi-scale features, which can emphasize features with high response to building semantic features at each scale, and weakening features with low response will enhance the representation of building features.

Table 3. Quantitative results on *Precision*, *Recall*, *F*1-*Score*, and *IoU* (in %) of different methods on the Inria Aerial Image Dataset. The best results are shown in bold.

Metrics	Methods	Austin	Chicago	Kitsap	Tyrol	Vienna	Average
Precision	FCN8s [37]	88.28	81.37	85.21	88.25	89.81	86.64
	U-Net [38]	89.92	**87.61**	84.03	87.62	89.65	87.77
	PSPNet [41]	84.58	80.57	81.01	85.57	87.47	83.84
	PANet [60]	87.72	77.13	80.68	86.26	84.89	83.34
	SiU-Net [19]	90.94	81.39	84.42	87.67	89.02	86.69
	BRRNet [27]	89.30	87.20	80.09	83.13	88.04	85.55
	AGPNet [25]	**91.72**	86.37	**85.91**	**90.30**	**91.45**	**89.15**
	Res2-Unet [65]	86.86	79.20	77.74	85.61	86.06	83.09
	FSIANet (Ours)	90.04	86.25	83.23	85.80	89.59	86.98
Recall	FCN8s [37]	87.32	**79.29**	70.41	80.89	83.39	80.26
	U-Net [38]	87.03	73.49	73.16	83.37	85.33	80.48
	PSPNet [41]	74.33	75.19	69.73	79.99	81.99	76.25
	PANet [60]	74.26	66.19	65.50	75.23	79.39	72.11
	SiU-Net [19]	86.39	78.27	73.55	82.27	84.60	81.02
	BRRNet [27]	89.07	75.78	77.57	85.85	85.44	82.74
	AGPNet [25]	86.81	78.69	76.24	82.71	85.11	81.91
	Res2-Unet [65]	84.70	78.06	72.40	83.09	84.90	80.63
	FSIANet (Ours)	**90.30**	78.75	**79.39**	**88.35**	**87.01**	**84.76**
*F*1-*Score*	FCN8s [37]	87.80	80.47	77.11	84.40	86.48	83.25
	U-Net [38]	88.45	79.94	78.22	85.44	87.43	83.90
	PSPNet [41]	79.12	77.79	74.95	82.69	84.64	79.84
	PANet [60]	80.43	71.24	72.30	80.37	82.04	77.28
	SiU-Net [19]	88.61	79.81	78.61	84.89	86.75	83.73
	BRRNet [27]	89.19	81.09	79.20	84.47	86.72	84.13
	AGPNet [25]	89.20	**82.35**	80.79	86.34	88.17	85.37
	Res2-Unet [65]	85.77	78.63	74.97	84.33	85.48	81.84
	FSIANet (Ours)	**90.17**	82.33	**81.26**	**87.06**	**88.28**	**85.82**
IoU	FCN8s [37]	78.25	67.32	62.74	73.02	76.18	71.50
	U-Net [38]	79.30	66.58	64.23	74.58	77.67	72.47
	PSPNet [41]	65.46	63.65	59.94	70.48	73.37	66.58
	PANet [60]	67.24	55.33	56.62	67.18	69.55	63.18
	SiU-Net [19]	79.54	66.39	64.76	73.74	76.61	72.21
	BRRNet [27]	80.48	68.19	65.57	73.11	76.58	72.79
	AGPNet [25]	80.50	**69.99**	67.77	75.96	78.84	74.61
	Res2-Unet [65]	75.09	64.78	59.96	72.90	74.64	69.47
	FSIANet (Ours)	**82.10**	69.97	**68.44**	**77.08**	**79.02**	**75.32**

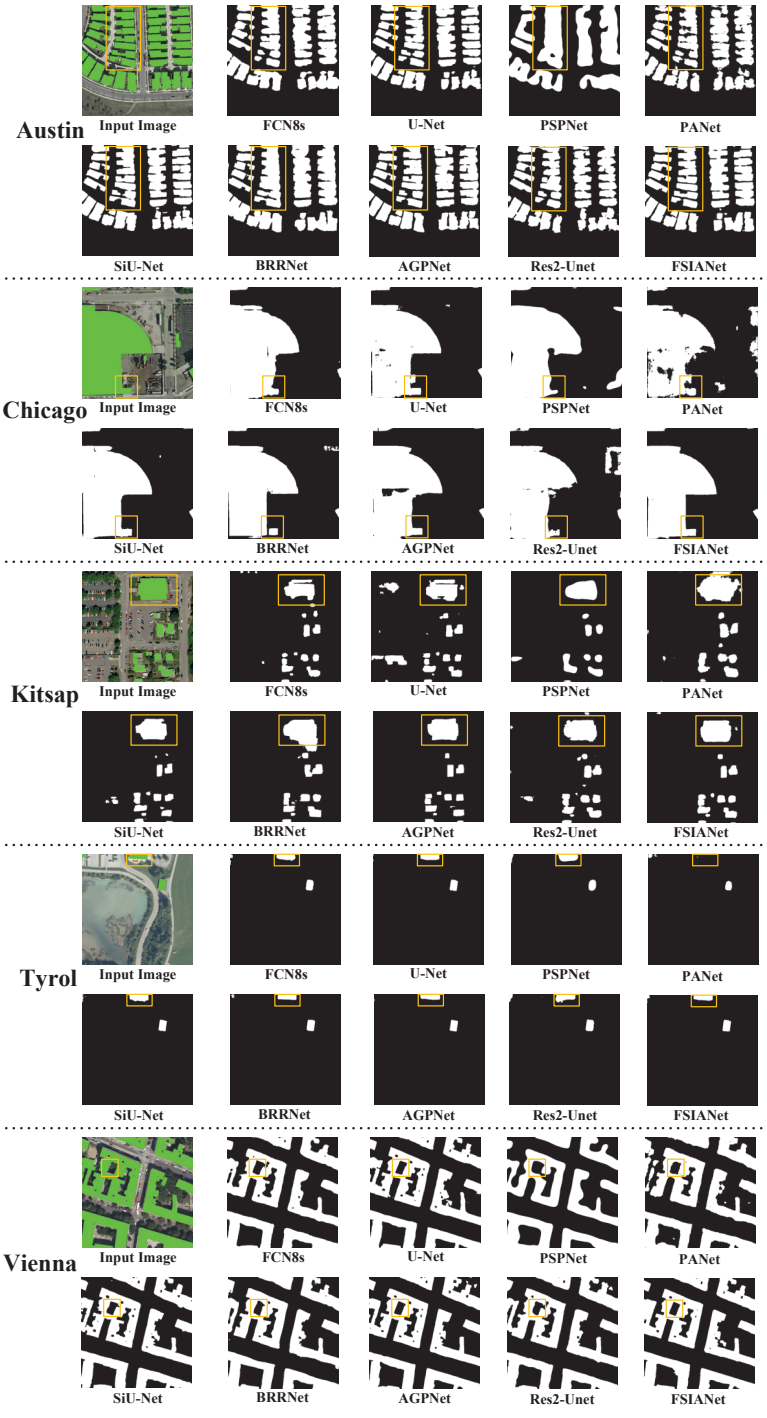


Figure 6. The visualization results of the proposed FSIANet and other comparison methods on the Inria Aerial Image Dataset.

4.4. Ablation Study

To further illustrate the effectiveness of our proposed innovations, ablation experiments on the East Asia Dataset are presented in Table 4. Specifically, the introduction of the FSIA shows much improvement in various indicators compared with only the backbone network. The FSIA module does not require learnable parameters, and the FSIA mechanism based on frequency domain information can effectively evaluate the informative abundance of feature maps and enhance feature representation by emphasizing more informative feature maps. After adding the ASPP, the performance of the network is not significantly improved or even slightly decreased. Therefore, our designed AFSAP in the network is able to mine multi-scale features, which can emphasize features with high response to building semantic features at each scale, while weakening features with low response can enhance the representation of building features.

In addition, we also implemented McNemar's test to further obviously verify the superiority of our method. Here, McNemar's test can be computed by Formula (17):

$$z = \frac{|N_{ij} - N_{ji}|}{\sqrt{N_{ij} + N_{ji}}} \tag{17}$$

where N_{ij} denotes the number of pixels that were correctly detected in method i but falsely detected in method j. For McNemar's test, $|z| > 1.96$ indicates a significant performance gap between the two methods [67]. McNemar's test of the ablation study on the East Asia Dataset is listed in Table 5. McNemar's test results present that the proposed method has a significant performance advantage after introducing FSIA and AFSAP.

Table 4. Ablation results on *Precision*, *Recall*, *F*1-*Score*, and *IoU* (in %) of our proposed FSIANet on the East Asia Dataset. The best results are shown in bold.

Methods	***Precision***	***Recall***	***F*1-*Score***	***IoU***
backbone	83.52	79.04	81.22	68.38
backbone+FSIA	84.27	79.29	81.71	69.07
backbone+FSIA+ASPP	**85.39**	78.62	81.86	69.30
backbone+FSIA+AFSAP (Full)	84.11	**80.75**	**82.39**	**70.06**

Table 5. McNemar's test of the ablation study over the proposed FSIANet on the East Asia Dataset.

FSIANet	**vs. Backbone**	**vs. Backbone+FSIA**	**vs. Backbone+FSIA+ASPP**
z value	154.26	80.27	28.58

Furthermore, to illustrate the rationale for the FSIANet design, the feature maps and discrete cosine transformation (DCT) results on the East Asia Dataset are shown in Figure 7. Here, we define an average frequency spectrum intensity (AFSI), which is the average of the frequency spectral values (computed by DCT) of a feature map. For AFSI, a higher value of AFSI means that building semantic and spatial information is more closely connected. Figure 7 mainly illustrates the visualization of the DCT in three channels of the feature map obtained from FSIANet. For example, in Figure 7(1-1–1-3), the more information the feature map carries, the bigger the corresponding AFSI is. This intuitively illustrates that FSIA can emphasize features with high response to building semantic features at each scale, and weakening features with low response will enhance the representation of building features.

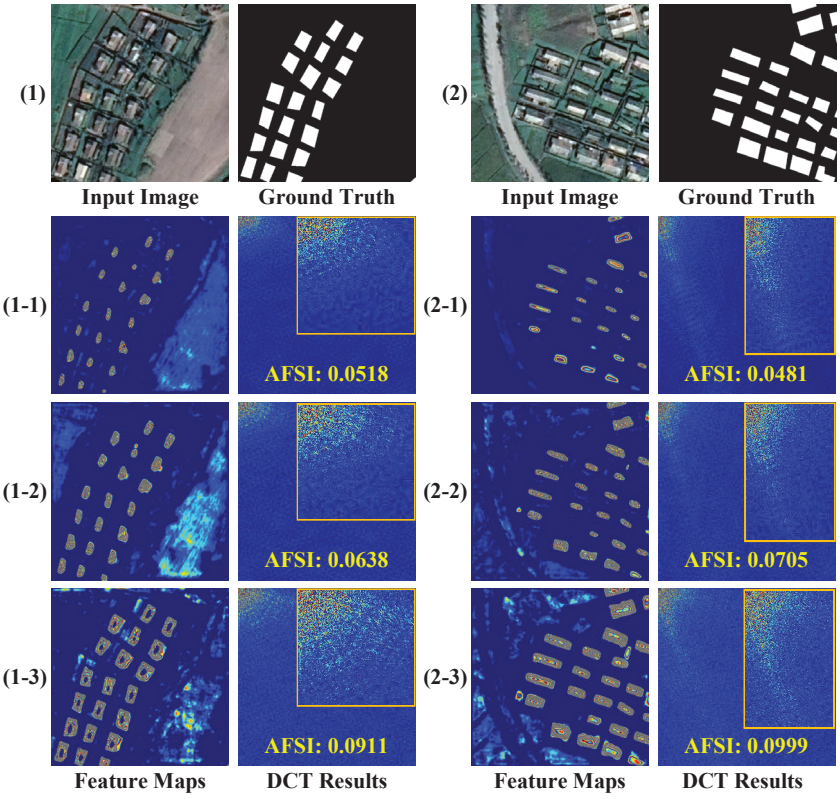


Figure 7. The feature maps and DCT results on the East Asia Dataset. Image (**1**): ((**1-1**)–(**1-3**)) represent different feature maps and their corresponding DCT results, respectively; image (**2**): ((**2-1**)–(**2-3**)) denote different feature maps and their corresponding DCT results, respectively.

5. Discussion

From the extensive experiments conducted above, it can be concluded that the proposed FSIA mechanism and AFSAP module can efficiently improve the performance of building extraction. In this section, these contributions are further discussed.

In FSIA, we utilize DCT to evaluate how informative a feature is, and reweight the features accordingly. Since its benefit has been confirmed in building extraction, it may potentially improve the performance of CNN-based methods over similar tasks such as change detection and road extraction, even more computer vision tasks. Considering that FSIA has no supervised parameters, it can be used in any CNN-based method without training. However, there are still several disadvantages to this distinctive attention mechanism. The most notable of them is that DCT can be time-consuming when processing feature maps with large spatial sizes. This problem can be further overcome in future work with a lightweight transformation.

6. Conclusions

In this work, efforts have been made to better tackle automatic building detection tasks in HR remote sensing data by proposing some computational-intelligence-based techniques. Namely, a classic encoder-decoder-like end-to-end deep convolutional neural network, FSIANet, with two newly proposed modules, FSIA and AFSAP, is exploited. The FSIA is able to mine useful information from the frequency spectrum of extracted features, thus improving the global feature representation of FSIANet. Notably, it does not need to be trained to acquire reliable ability, which is different from most of the other

attention mechanisms. In addition, the ASPP-inspired feature pyramid, AFSAP, is utilized to promote the detection of building-like objects. Compared to ASPP, the AFSAP can achieve more pronounced performance improvement with the help of FSIA. As a result, the proposed FSIANet has successfully outperformed several newly proposed cutting-edge deep-learning-based methods in two widely used large-scale HR remote sensing building detection datasets. For future work, more efforts can be made to expand the usage of frequency–domain-based analysis in the deep-learning-based methods, which have the potential to facilitate finer annotation of buildings in complicated scenes.

Author Contributions: Conceptualization, D.F.; methodology, D.F.; validation, H.C.; investigation, L.Z.; writing—original draft preparation, D.F.; writing—review and editing, H.C. and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62102314, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grants 2021JQ-721, 2021JQ-708, and 2022JQ-635, and in part by the Special Scientific Research Projects of Shaanxi Provincial Department of Education 20JK0918.

Data Availability Statement: Not applicable.

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.

Abbreviations

AFSAP	Atrous Frequency Spectrum Attention Pyramid
ASPP	Atrous Spatial Pyramid Pooling
BRRNet	Building Residual Refine Network
CNN	Convolutional Neural Network
DCT	Discrete Cosine Transformation
FCN	Fully Convolutional Network
FSIANet	Frequency Spectrum Intensity Attention Network
HR	High-Resolution
SOTA	State-of-the-Art
AFSI	Average Frequency Spectrum Intensity

References

1. Wu, Y.; Ma, W.; Gong, M.; Su, L.; Jiao, L. A novel point-matching algorithm based on fast sample consensus for image registration. *IEEE Geosci. Remote Sens. Lett.* **2014**, *12*, 43–47. [CrossRef]
2. Wu, Y.; Li, J.; Yuan, Y.; Qin, A.; Miao, Q.G.; Gong, M.G. Commonality autoencoder: Learning common features for change detection from heterogeneous images. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**. [CrossRef] [PubMed]
3. Li, J.; Li, H.; Liu, Y.; Gong, M. Multi-fidelity evolutionary multitasking optimization for hyperspectral endmember extraction. *Appl. Soft Comput.* **2021**, *111*, 107713. [CrossRef]
4. Lv, Z.; Li, G.; Jin, Z.; Benediktsson, J.A.; Foody, G.M. Iterative training sample expansion to increase and balance the accuracy of land classification from VHR imagery. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 139–150. [CrossRef]
5. Zhang, M.; Gong, M.; Mao, Y.; Li, J.; Wu, Y. Unsupervised feature extraction in hyperspectral images based on wasserstein generative adversarial network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 2669–2688. [CrossRef]
6. Zhang, M.; Gong, M.; He, H.; Zhu, S. Symmetric all convolutional neural-network-based unsupervised feature extraction for hyperspectral images classification. *IEEE Trans. Cybern.* **2020**. [CrossRef]
7. Lv, Z.; Wang, F.; Cui, G.; Benediktsson, J.A.; Lei, T.; Sun, W. Spatial–Spectral Attention Network Guided With Change Magnitude Image for Land Cover Change Detection Using Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–12. [CrossRef]
8. Gong, M.; Jiang, F.; Qin, A.K.; Liu, T.; Zhan, T.; Lu, D.; Zheng, H.; Zhang, M. A Spectral and Spatial Attention Network for Change Detection in Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–14. [CrossRef]
9. Wang, Z.; Jiang, F.; Liu, T.; Xie, F.; Li, P. Attention-Based Spatial and Spectral Network with PCA-Guided Self-Supervised Feature Extraction for Change Detection in Hyperspectral Images. *Remote Sens.* **2021**, *13*, 4927. [CrossRef]

10. Shivappriya, S.; Priyadarsini, M.J.P.; Stateczny, A.; Puttamadappa, C.; Parameshachari, B. Cascade object detection and remote sensing object detection method based on trainable activation function. *Remote Sens.* **2021**, *13*, 200. [CrossRef]
11. Ghanea, M.; Moallem, P.; Momeni, M. Building extraction from high-resolution satellite images in urban areas: Recent methods and strategies against significant challenges. *Int. J. Remote Sens.* **2016**, *37*, 5234–5248. [CrossRef]
12. Singh, D.; Kaur, M.; Jabarulla, M.Y.; Kumar, V.; Lee, H.N. Evolving fusion-based visibility restoration model for hazy remote sensing images using dynamic differential evolution. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
13. Wu, Y.; Zhang, Y.; Fan, X.; Gong, M.; Miao, Q.; Ma, W. INENet: Inliers Estimation Network with Similarity Learning for Partial Overlapping Registration. *IEEE Trans. Circuits Syst. Video Technol.* **2022**. [CrossRef]
14. Liu, T.; Gong, M.; Jiang, F.; Zhang, Y.; Li, H. Landslide Inventory Mapping Method Based on Adaptive Histogram-Mean Distance with Bitemporal VHR Aerial Images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
15. Wu, Y.; Liu, Y.; Gong, M.; Gong, P.; Li, H.; Tang, Z.; Miao, Q.; Ma, W. Multi-View Point Cloud Registration Based on Evolutionary Multitasking With Bi-Channel Knowledge Sharing Mechanism. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**. [CrossRef]
16. Awrangjeb, M.; Lu, G.; Fraser, C. Automatic building extraction from LiDAR data covering complex urban scenes. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *40*, 25. [CrossRef]
17. Lv, Z.; Liu, T.; Benediktsson, J.A.; Falco, N. Land cover change detection techniques: Very-high-resolution optical images: A review. *IEEE Geosci. Remote Sens. Mag.* **2021**, *10*, 44–63. [CrossRef]
18. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Can semantic labeling methods generalize to any city? The inria aerial image labeling benchmark. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; pp. 3226–3229.
19. 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]
20. Liu, T.; Gong, M.; Lu, D.; Zhang, Q.; Zheng, H.; Jiang, F.; Zhang, M. Building Change Detection for VHR Remote Sensing Images via Local–Global Pyramid Network and Cross-Task Transfer Learning Strategy. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–17. [CrossRef]
21. Bi, Q.; Qin, K.; Zhang, H.; Zhang, Y.; Li, Z.; Xu, K. A multi-scale filtering building index for building extraction in very high-resolution satellite imagery. *Remote Sens.* **2019**, *11*, 482. [CrossRef]
22. Ma, J.; Wu, L.; Tang, X.; Liu, F.; Zhang, X.; Jiao, L. Building extraction of aerial images by a global and multi-scale encoder-decoder network. *Remote Sens.* **2020**, *12*, 2350. [CrossRef]
23. Xia, L.; Zhang, X.; Zhang, J.; Yang, H.; Chen, T. Building extraction from very-high-resolution remote sensing images using semi-supervised semantic edge detection. *Remote Sens.* **2021**, *13*, 2187. [CrossRef]
24. Liao, C.; Hu, H.; Li, H.; Ge, X.; Chen, M.; Li, C.; Zhu, Q. Joint learning of contour and structure for boundary-preserved building extraction. *Remote Sens.* **2021**, *13*, 1049. [CrossRef]
25. Deng, W.; Shi, Q.; Li, J. Attention-gate-based encoder–decoder network for automatical building extraction. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 2611–2620. [CrossRef]
26. Zhao, H.; Zhang, H.; Zheng, X. A Multiscale Attention-Guided UNet++ with Edge Constraint for Building Extraction from High Spatial Resolution Imagery. *Applied Sci.* **2022**, *12*, 5960. [CrossRef]
27. Shao, Z.; Tang, P.; Wang, Z.; Saleem, N.; Yam, S.; Sommai, C. BRRNet: A fully convolutional neural network for automatic building extraction from high-resolution remote sensing images. *Remote Sens.* **2020**, *12*, 1050. [CrossRef]
28. Yang, G.; Zhang, Q.; Zhang, G. EANet: Edge-aware network for the extraction of buildings from aerial images. *Remote Sens.* **2020**, *12*, 2161. [CrossRef]
29. Zheng, H.; Gong, M.; Liu, T.; Jiang, F.; Zhan, T.; Lu, D.; Zhang, M. HFA-Net: High frequency attention siamese network for building change detection in VHR remote sensing images. *Pattern Recognit.* **2022**, *129*, 108717. [CrossRef]
30. He, H.; Yang, D.; Wang, S.; Wang, S.; Li, Y. Road extraction by using atrous spatial pyramid pooling integrated encoder-decoder network and structural similarity loss. *Remote Sens.* **2019**, *11*, 1015. [CrossRef]
31. Yu, M.; Zhang, W.; Chen, X.; Liu, Y.; Niu, J. An End-to-End Atrous Spatial Pyramid Pooling and Skip-Connections Generative Adversarial Segmentation Network for Building Extraction from High-Resolution Aerial Images. *Appl. Sci.* **2022**, *12*, 5151. [CrossRef]
32. Zhang, L.; Huang, X.; Huang, B.; Li, P. A pixel shape index coupled with spectral information for classification of high spatial resolution remotely sensed imagery. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 2950–2961. [CrossRef]
33. Mongus, D.; Lukač, N.; Žalik, B. Ground and building extraction from LiDAR data based on differential morphological profiles and locally fitted surfaces. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 145–156. [CrossRef]
34. Huang, X.; Zhang, L. A multidirectional and multiscale morphological index for automatic building extraction from multispectral GeoEye-1 imagery. *Photogramm. Eng. Remote Sens.* **2011**, *77*, 721–732. [CrossRef]
35. Huang, X.; Zhang, L.; Zhu, T. Building change detection from multitemporal high-resolution remotely sensed images based on a morphological building index. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *7*, 105–115. [CrossRef]
36. You, Y.; Wang, S.; Ma, Y.; Chen, G.; Wang, B.; Shen, M.; Liu, W. Building detection from VHR remote sensing imagery based on the morphological building index. *Remote Sens.* **2018**, *10*, 1287. [CrossRef]
37. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.

38. 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; Springer: Berlin, Germany, 2015; pp. 234–241.
39. 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]
40. Zhang, Y.; Gong, M.; Li, J.; Zhang, M.; Jiang, F.; Zhao, H. Self-Supervised Monocular Depth Estimation with Multiscale Perception. *IEEE Trans. Image Process.* **2022**, *31*, 3251–3266. [CrossRef]
41. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
42. 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.* **2017**, *40*, 834–848. [CrossRef]
43. Luo, L.; Li, P.; Yan, X. Deep learning-based building extraction from remote sensing images: A comprehensive review. *Energies* **2021**, *14*, 7982. [CrossRef]
44. Wang, L.; Fang, S.; Meng, X.; Li, R. Building extraction with vision transformer. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [CrossRef]
45. Wang, Y.; Liang, B.; Ding, M.; Li, J. Dense semantic labeling with atrous spatial pyramid pooling and decoder for high-resolution remote sensing imagery. *Remote Sens.* **2018**, *11*, 20. [CrossRef]
46. Weihong, C.; Baoyu, X.; Liyao, Z. Multi-scale fully convolutional neural network for building extraction. *Acta Geodaetica et Cartogr. Sinica* **2019**, *48*, 597.
47. Yuan, W.; Xu, W. MSST-Net: A Multi-Scale Adaptive Network for Building Extraction from Remote Sensing Images Based on Swin Transformer. *Remote Sens.* **2021**, *13*, 4743. [CrossRef]
48. Qiu, Y.; Wu, F.; Yin, J.; Liu, C.; Gong, X.; Wang, A. MSL-Net: An Efficient Network for Building Extraction from Aerial Imagery. *Remote Sens.* **2022**, *14*, 3914. [CrossRef]
49. 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 Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
50. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
51. Gong, M.; Li, J.; Zhang, Y.; Wu, Y.; Zhang, M. Two-Path Aggregation Attention Network with Quad-Patch Data Augmentation for Few-shot Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**. [CrossRef]
52. Guo, M.H.; Xu, T.X.; Liu, J.J.; Liu, Z.N.; Jiang, P.T.; Mu, T.J.; Zhang, S.H.; Martin, R.R.; Cheng, M.M.; Hu, S.M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, 1–38. [CrossRef]
53. Ghaffarian, S.; Valente, J.; Van Der Voort, M.; Tekinerdogan, B. Effect of attention mechanism in deep learning-based remote sensing image processing: A systematic literature review. *Remote Sens.* **2021**, *13*, 2965. [CrossRef]
54. 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.
55. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image super-resolution using very deep residual channel attention networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 286–301.
56. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13–19.
57. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
58. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794–7803.
59. Pan, X.; Yang, F.; Gao, L.; Chen, Z.; Zhang, B.; Fan, H.; Ren, J. Building extraction from high-resolution aerial imagery using a generative adversarial network with spatial and channel attention mechanisms. *Remote Sens.* **2019**, *11*, 917. [CrossRef]
60. Li, H.; Xiong, P.; An, J.; Wang, L. Pyramid attention network for semantic segmentation. *arXiv* **2018**, arXiv:1805.10180.
61. Guo, H.; Shi, Q.; Du, B.; Zhang, L.; Wang, D.; Ding, H. Scene-driven multitask parallel attention network for building extraction in high-resolution remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 4287–4306. [CrossRef]
62. Ye, Z.; Fu, Y.; Gan, M.; Deng, J.; Comber, A.; Wang, K. Building extraction from very high resolution aerial imagery using joint attention deep neural network. *Remote Sens.* **2019**, *11*, 2970. [CrossRef]
63. Guo, M.; Liu, H.; Xu, Y.; Huang, Y. Building extraction based on U-Net with an attention block and multiple losses. *Remote Sens.* **2020**, *12*, 1400. [CrossRef]
64. Chen, X.; Qiu, C.; Guo, W.; Yu, A.; Tong, X.; Schmitt, M. Multiscale feature learning by transformer for building extraction from satellite images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
65. Chen, F.; Wang, N.; Yu, B.; Wang, L. Res2-Unet, a New Deep Architecture for Building Detection from High Spatial Resolution Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 1494–1501. [CrossRef]

66. Guo, H.; Du, B.; Zhang, L.; Su, X. A coarse-to-fine boundary refinement network for building footprint extraction from remote sensing imagery. *ISPRS J. Photogramm. Remote Sens.* **2022**, *183*, 240–252. [CrossRef]
67. Foody, G.M. Thematic map comparison. *Photogramm. Eng. Remote Sens.* **2004**, *70*, 627–633. [CrossRef]

remote sensing

MDPI

Article

ELCD: Efficient Lunar Crater Detection Based on Attention Mechanisms and Multiscale Feature Fusion Networks from Digital Elevation Models

Lili Fan *, Jiabin Yuan, Keke Zha and Xunan Wang

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
* Correspondence: fanlily913@nuaa.edu.cn

Abstract: The detection and counting of lunar impact craters are crucial for the selection of detector landing sites and the estimation of the age of the Moon. However, traditional crater detection methods are based on machine learning and image processing technologies. These are inefficient for situations with different distributions, overlaps, and crater sizes, and most of them mainly focus on the accuracy of detection and ignore the efficiency. In this paper, we propose an efficient lunar crater detection (ELCD) algorithm based on a novel crater edge segmentation network (AFNet) to detect lunar craters from digital elevation model (DEM) data. First, in AFNet, a lightweight attention mechanism module is introduced to enhance the feature extract capabilities of networks, and a new multiscale feature fusion module is designed by fusing different multi-level feature maps to reduce the information loss of the output map. Then, considering the imbalance in the classification and the distributions of the crater data, an efficient crater edge segmentation loss function (CESL) is designed to improve the network optimization performance. Lastly, the crater positions are obtained from the network output map by the crater edge extraction (CEA) algorithm. The experiment was conducted on the PyTorch platform using two lunar crater catalogs to evaluate the ELCD. The experimental results show that ELCD has a superior detection accuracy and inference speed compared with other state-of-the-art crater detection algorithms. As with most crater detection models that use DEM data, some small craters may be considered to be noise that cannot be detected. The proposed algorithm can be used to improve the accuracy and speed of deep space probes in detecting candidate landing sites, and the discovery of new craters can increase the size of the original data set.

Keywords: crater detection; image segmentation; moon; deep learning; remote sensing

Citation: Fan, L.; Yuan, J.; Zha, K.; Wang, X. ELCD: Efficient Lunar Crater Detection Based on Attention Mechanisms and Multiscale Feature Fusion Networks from Digital Elevation Models. *Remote Sens.* **2022**, *14*, 5225. https://doi.org/10.3390/rs14205225

Academic Editors: Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Received: 22 September 2022
Accepted: 17 October 2022
Published: 19 October 2022

1. Introduction

Impact craters constitute an important property of the lunar surface. Impact craters provide significant information for lunar evolution [1,2]. For example, the distribution and number of craters are often used to estimate the relative age of the Moon [3–5], and craters also provide important landmark information to accurately guide spacecraft to land [6,7]. The discovery of impact craters on the lunar surface is very important for studying the Moon, for example, by using the manual analysis and comparative evaluation of craters' images with different features to identify the permanently shadowed lunar polar regions [8]. In the study of crater counting, some crater catalogs have been formed manually by planetary scientists, such as the crater catalog of the Moon (diameter 5∼20 km [9], diameter $\geq$ 20 km [10]). However, the manual discovery of craters is time-consuming and laborious, and because experts may disagree on the interpretation of image data, the manual marking of craters also faces consistency and repeatability challenges.

Several automatic crater detection algorithms have been proposed to detect craters, and these can be roughly grouped into two categories. The first kind of method is unsupervised-based algorithms, which use digital image processing technology to detect craters, and the

second kind of method is supervised-based algorithms, which employ machine learning or deep learning to extract impact craters.

Unsupervised-based automatic crater feature extraction algorithms are mainly based on traditional image processing methods, including Hough transforms [11–13], template matching [14], edge detection, convex grouping [15], and other recognition techniques. For example, the performance of a Hough transform applied to large scale crater counting was evaluated [13] in terms of its ability to automatically detect craters down to sub-km sizes on high-resolution images of the Martian surface. The Canny edge detector is widely used in computer vision to locate sharp intensity changes and to find object boundaries in an image. The combined adaptive Canny algorithm, which uses histograms of images and multi-scale Gaussian filtering, was used in [16] to achieve a crater matching rate of better than 85%. However, for irregular, incomplete shapes and areas with a high degree of overlap, the detection accuracy of such methods is poor. Furthermore, Chen et al. [17] used terrain analysis and mathematical morphology methods to identify different types of impact craters, which fit the crater edge based on the Moon's digital elevation model (DEM) data. In contrast, the mathematical fitting method is more reliable than the Hough ring transform algorithm, but its computational complexity is higher for the identification of large, dense craters.

Automatic crater supervised-based technology has developed rapidly through machine learning and deep learning methods. Machine learning-based methods often involve building a classifier to recognize candidate craters, and common classifiers, such as the principal components analysis [18], decision trees technique [19], support vector machine, and other hybrid methods [20], are used to classify candidate craters. To improve the classification accuracy of small craters, Kang et al. [21] combined a histogram of oriented gradient features and the support vector machine classifier to extract small-scale impact craters from charge-coupled device images. Furthermore, based on the scale of training samples generated from the surface imagery and digital elevation models of the Moon, [22] proposed an active machine learning approach to automatically detect candidate craters by training a classifier with better performance. These methods are able to recognize craters or non-craters with a high classification accuracy. However, they need to extract features manually when training a classifier to detect craters. For large-scale and high-density crater detection, most of them have poor recognition accuracy and robustness. Some of them cannot count craters or locate the positions of craters.

Deep learning, especially when based on convolution neural networks (CNNs), has achieved great success in solving problems with image classification, image segmentation [23,24], and synthetic aperture radar (SAR) automatic object detection [25,26] in the remote sensing fields. The CNN is a key representative network structure in deep learning techniques. Such techniques are different from machine learning techniques, which are more efficient and portable without a set of human-designed features [27]. Impact crater detection based on deep learning is an important method in the vision-based navigation systems and is used to solve the task of pinpoint landing on the Moon. Some works [28,29] have used CNN feature extraction and standard image processing technologies to detect and match the observed craters, which were used as visual landmark measurements by the navigation filter. Moreover, image segmentation [23,30] and object detection methods based on CNNs, e.g., faster region-CNN (R-CNN) [31] and mask R-CNN [32], are used to solve crater detection problems. For example, Tewari et al. [32] utilized the mask R-CNN framework to detect craters from optical images, digital elevation maps, and slope maps by post-processing to eliminate duplicate craters and extract the craters' global locations. Moreover, to improve the detection accuracy of small-impact craters, [33] proposed an end-to-end high-resolution feature pyramid network framework, denoted as HRFPNet. HRFPNet uses a new backbone with a feature aggregation module to enhance the feature extraction capability of small craters from thermal infrared imaging on Mars. However, most object detection-based methods need to consider the generation of the number of candidate boxes. For highly overlapping and dense craters, the quality of the generation of

a large number of duplicate bounding boxes may affect the recognition speed and accuracy of crater detection. Therefore, most object detection schemes display relatively poor performances and high levels of computational complexity in crater detection.

Crater detection is also solved as a semantic segmentation problem, in which the rims or edges of craters can be extracted by pixel-level classification, and the crater position and size can be obtained by a post-pipeline method. For example, a semantic segmentation method based on the fully convolutional neural network was proposed [34]. This method uses different feature maps with multi-scale receptive fields to detect multi-scale impact craters from remotely sensed planetary images. Moreover, semantic segmentation models [35–37] based on U-net [38] have been presented to detect craters. Silburtet et al. [35] proposed DeepMoon based on the U-net network structure to recognize lunar craters from DEM data. This method can successfully identify about 45% of newly discovered craters in its validation data. However, the U-net network structure loses large amounts of detailed information in the encoder of the network, which leads to poor crater image contour recovery in the decoder process. To improve the accuracy of crater detection, a new network structure, ERU-Net [36], introduced the deep residual network module to improve the crater feature extraction ability. This successfully achieved a recall rate of 81.2% and a precision rate of 75.4% in lunar crater recognition when training 30,000 DEM data images. Furthermore, to explore craters on Mars, DeLatte et al. [37] employed segmentation convolution neural networks based on U-net for automatic crater detection from Martian daytime infra-red images. This method identified 65–75% of craters in common with a human-annotated dataset, and [39] used the ResUNET [30] model to detect craters with the global maps and infra-red imagery for Mars. However, resources in the deep space environment are limited [40]; thus, automated crater detection methods require a balance between model computational complexity and identification efficiency. Most of the above methods ignore the computational complexity of the model.

The deep learning-based algorithms described above have different improved optimization approaches for different crater tasks. However, the majority of object detection schemes perform relatively poorly as they are constrained by their vanilla network architectures or semantic segmentation. By comparing the network complexity and recognition results, it can be seen that crater detection methods based on the semantic segmentation model are more efficient than the end-to-end object detection model. However, most semantic segmentation-based crater detection methods mainly focus on the accuracy of recognition and neglect the reasoning speed of the network. Moreover, due to crater images having different distributions, degrees of overlap, and sizes on the surface of the Moon, and because the crater data may be imbalanced, crater detection algorithms based on semantic segmentation networks may suffer from significant performance degradation. Therefore, achieving a fast and effective crater detection method with a high level of precision based on a semantic segmentation model represents a challenging scenario.

To address this issue, in this study, we establish an efficient lunar crater detection (ELCD) algorithm that addresses the requirements for accurate and fast crater detection. In the ELCD algorithm, first, the crater edge is segmented by the attention mechanisms and multiscale feature fusion networks (AFNet). Then, the crater position and size are extracted by postprocessing based on the crater extract algorithm (CEA). In AFNet, a light-weight attention mechanism is used to improve the feature extraction ability of the network, and a new multiscale feature fusion (MFF) module is designed in the upsampling process of the network to reduce the loss of detail in the semantic segmentation results. In addition, we consider the crater data imbalance of the classification and distributions and design a new crater edge segmentation loss (CESL) function for network training. The proposed loss function improves the optimization ability and convergence speed of the network through adaptive balance weights.

The main contributions of the paper are as follows:

- We propose an efficient crater detection network based on a new semantic segmentation network architecture, AFNet, which uses the lightweight attention mechanism and multiscale feature fusion module to provide better and faster detection of lunar impact craters.
- To improve the optimization capability of the network, we present the crater edge segmentation loss function, which considers the imbalance of classification and distributions of crater data to calculate the loss value using the different degrees of imbalance in the data.
- The experiment is conducted on the PyTorch platform [41] with lunar DEM data to verify the effectiveness of the ELCD. The results show that the ELCD outperforms the state-of-the-art crater detection models in terms of its detection accuracy and inference speed.

The rest of this paper is organized as follows: Section 2 describes the proposed network architecture, the design of the crater edge segmentation loss function, the crater edge extraction algorithm, and the details of the experiment. Section 3 provides the experimental results, and Section 4 presents our discussion. Eventually, in Section 5, we conclude our work.

2. Materials and Methods

The workflow description of two stages of the lunar crater detection method using DEM data is shown in Figure 1. The workflow includes two parts: (i) crater edge prediction by the semantic segmentation network AFNet and (ii) crater edge extraction with the post-pipeline method. The details of the ELCD are as follows. The workflow input is the lunar crater DEM image. The DEM contains abundant 3D morphology and topography morphological characteristics, and it is insensitive to light [27]. The workflow output is the crater's positional information, such as its longitude, latitude, and radius, which is determined by the crater edge extraction algorithm. First, crater images with different degrees of size, overlap, and distribution are transferred to the crater edge segmentation network to undergo crater edge prediction. Then, the network prediction results are processed with a post-processing pipeline based on the match template method to obtain the location information and radial size of craters.

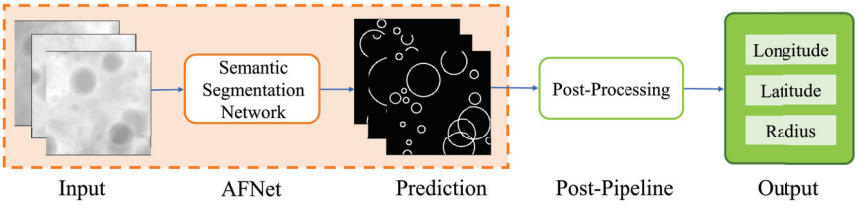


Figure 1. Workflow of two stages used in the crater detection method based on the semantic segmentation network and the crater edge extraction method. The network input is the DEM image. The digital elevation model (DEM) image is first processed by AFNet to recognize crater edges by pixel-level classification. Then, the prediction result of the crater images from network training is processed by a post-processing pipeline based on the match template method to detect the location information and radius size of craters.

2.1. AFNet

To obtain efficient crater edge prediction results, we formally describe the crater edge detection network architecture, as shown in Figure 2. The AFNet includes three parts: the network encoder, feature fusion, and decoder. In Figure 2, the black line is the network encoder, the blue line denotes the process of feature fusion, and the orange line represents the network decoder process. The network input is the gray DEM image, which has

a fixed size of 256 × 256 pixels, and the output is the pixel-level classification for the prediction result.

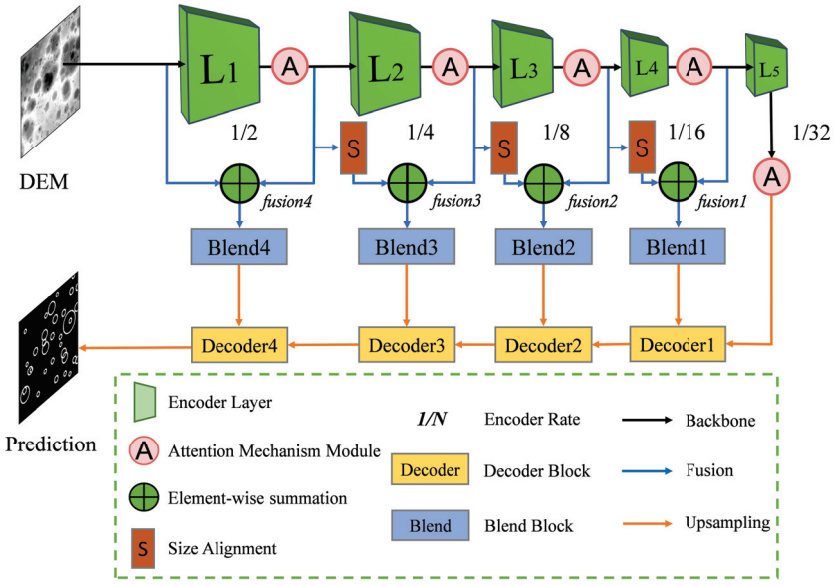


Figure 2. AFNet framework based on the improved VGG-16. The input is the DEM image transferred to the network encoder process (green trapezoid). First, the DEM image is processed with a $1/N$ downsampling rate with an attention mechanism module (pink circle) and five convolution blocks. Then, feature maps with different resolutions are saved and fused by the multiscale feature fusion module (blue line) with element-wise summation (green $\oplus$) and the data blending block (blue squares) through the decoder process (yellow squares) to get a more fine-grained output feature map. The final output result denotes the network prediction results with pixel-level classification.

In encoder processing, we use the visual geometry group-16 (VGG-16) [42] as the backbone to extract the crater features. This allows us to obtain a bigger receptive field using fewer parameters compared with other network structures. The backbone network includes five feature extraction blocks, denoted as $L = \{L_1, L_2, \ldots, L_i\}$, where i is the number of feature extraction blocks. At the end of each feature extract block, we introduce the attention mechanism module to extract the important features of the crater. In L_1 and L_2, each feature extraction block contains two convolution layers: an attention machine module and a max-pooling layer. L_3, L_4 and L_5 contains three convolution layers, an attention machine module, and a max-pooling layer, and all convolution layers use a 3×3 convolution kernel in each block.

In feature fusion, to obtain a more fine-grained feature map in the network decoder, we designed a simple and efficient MFF to obtain more fine-grained output feature maps. The four fusion modules $fusion_j$ are shown in Figure 2 and $j = \{1, 2, 3, 4\}$. The MFF first uses the element wisdom summation (green in Figure 2) to fuse a low-resolution feature map and a high-resolution feature map in each step of the upsampling process (decoder). Then, the obtained fusion feature map is blended and transferred to the decoder process as an input for the next step (blue squares in Figure 2).

In decoder processing, the bilinear interpolation operation is used to restore the size of the feature maps by four decoder blocks, $Decoder_k$, $k = \{1, 2, 3, 4\}$ (yellow squares in Figure 2). We use 2× upsampling and fuse more rice feature map information in each decoder to restore the feature map to its original size.

2.2. Attention Mechanism Module

The original impact craters have different density distributions, sizes, and degrees of overlap in the different lunar regions. A description of the characteristics of crater DEM data used in network training is given in Figure 3. When the crater DEM images are processed by random clipping, they may have an incomplete shape. These crater characteristics bring performance challenges to the semantic segmentation network.

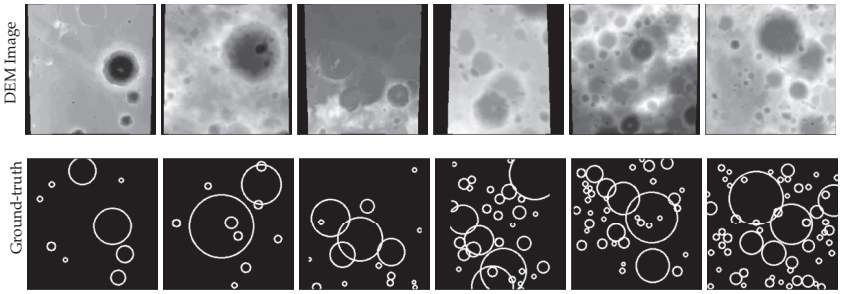


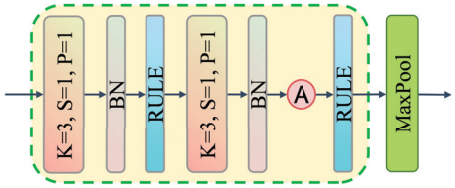
Figure 3. The different characteristics of craters on the surface of the Moon from DEM data. The top figure is the original crater images, and the bottom figure denotes the labeled images denoted as the ground truth. (**a**–**c**) show the distribution of sparse craters, (**d**–**f**) denote the distribution of dense craters, (**b**–**f**) represent the different degrees of crater overlap, (**c**–**f**) show the incomplete craters.

In the encoder, to improve the feature extraction ability of the network, we introduce the attention mechanism through efficient channel attention (ECA) [43], which is attached to the end of each feature extraction block of the proposed network to enhance the extraction of important features. Efficient channel attention with the lightweight module has great potential to produce a trade-off between performance and complexity. This only involves a handful of parameters while bringing a clear performance gain. The ECA block is termed an attention mechanism, as shown in Figure 2 with a pink circle. In the ECA, 1D convolution with a kernel size of 3 was used to achieve information exchange between channels. The details of the ECA block attached to the end of the five feature blocks are given in Figure 4. The ECA module was placed behind the activate function rectified linear unit (ReLU) in each feature extraction block. Figure 4a denotes the location of the ECA in the feature extraction block $\{L_1, L_2\}$, and Figure 4b shows the location of the ECA in the feature extraction block $\{L_3, L_4, L_5\}$ in the decoder process of the network. The ECA can combine the crater channel and spatial attention to enhance crater feature aggregation, which can enhance the extraction of salient crater features.

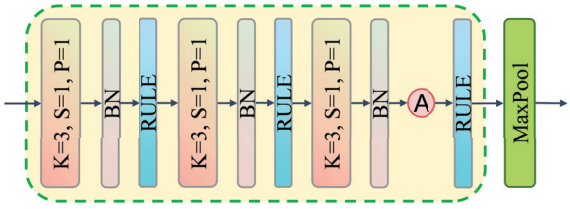
2.3. Multiscale Feature Fusion Module

Visual features with a coarse spatial resolution can be obtained by the encoder process. During the network encoder process, shallow crater networks can learn some local features because of the low perception threshold, and the deeper convolution layer can obtain more abstract features. With the deepening of the network, the receptive field of the network becomes larger, but because of the down-sampling operation, a great deal of detailed information may be lost. The purpose of the decoder process is to obtain a segmented prediction image with the same input size through the upsampling operation. Traditional segmentation networks use the simple upsampling module with skipped lateral connections to restore the feature map, which may cause the restored feature map to lack detailed features. To overcome the problem of poor image contour recovery in the decoder process, we designed a simple and efficient multiscale feature fusion module to fuse more low-layer features in each decoder block. The four multiscale feature fusion modules $fusion_j$, $j = \{1, 2, 3, 4\}$, are shown in Figure 5. We first obtained feature maps of different resolutions from the network encoder process. Then, we fused two close feature maps as

the upsampling input for the MFF to obtain an output feature map with more fine-grained information.

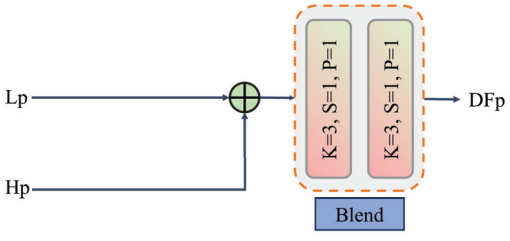


(**a**) ECA in L_1, L_2

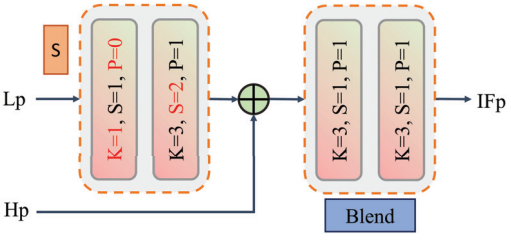


(**b**) ECA in L_3, L_4, L_5

Figure 4. ECA module. (**a**) ECA module in L_1 and L_2, (**b**) ECA module in L_3, L_4, and L_5.



(**a**) Direction Fusion



(**b**) Indirection Fusion

Figure 5. Multiscale feature fusion module. (**a**) is the direct fusion with the same feature map resolution and size. (**b**) denotes indirect fusion with feature maps of two different resolutions.

The MFF included two cases, direct fusion and indirect fusion, as shown in Figure 5. Figure 5a shows direct fusion for two feature maps of the same resolution: $fusion_1 - fusion_3$. Figure 5b denotes indirect fusion with two different resolution feature maps: $fusion_4$. In direct fusion, a low-resolution feature map denoted as Lp and high-resolution feature map represented as Hp have the same resolution. They use direct fusion by the element summation operation to obtain the fusion feature map. However, the size of the feature map is often different and usually has a two-fold difference in size after the encoder process. Therefore, processing is done through the indirect fusion module. In the indirect fusion module, the low-resolution feature map is not the same as the high-resolution feature

map. Lp is first processed with the size alignment module to obtain the same resolution as Hp. Then, the two maps are fused by element summation to obtain the fusion feature map. Finally, the fusion feature map is transferred to the blending block (blue squares in Figure 2) to obtain the final fusion output, the indirect fusion feature map denoted as IFp or the direct fusion feature map indicated as DFp, as the branch input for upsampling processing.

The size alignment module includes a 1×1 convolution kernel to reduce the dimensions and a 3×3 convolution kernel. This stride is set to 2 to adjust the map to the same size as Hp. The blend module contains the simple two 3×3 convolution kernel network to blend the fusion results. The final fusion feature maps have richer low-layer features, which could help us to obtain high-quality output prediction results in the encoder process.

2.4. Crater Edge Segmentation Loss Function

In the crater prediction network, crater images can be divided into foreground images and background images by pixel-level segmentation. In a crater image, the foreground image is the segmented object (crater edge), and the background image represents everything but the object. However, most crater detection methods based on segmentation networks use traditional loss functions, such as the cross-entropy (CE) loss function [35–37], to train the network, and they cannot overcome the variation in size and the serious crater data imbalance problem, resulting in a performance decrease. The CE can be computed as

$$CE(p_i, y_i) = \begin{cases} -log(p_i), & y_i = 1 \\ -log(1 - p_i), & otherwize \end{cases} \tag{1}$$

where p_i is prediction value of the network, y_i is the ground-truth, and $p_i \in [0,1]$, y$\in \{0,1\}$.

However, in cross-entropy loss, the weight of each sample is the same, and the CE loss is overwhelmed when facing the data classification imbalance. Later, the focal loss (FL) function [44] considering the classification imbalance in dense object detection was proposed to improve the network performance. The FL is defined as

$$FL(p_i) = -\alpha(1 - p_i)^{\gamma} log(p_i) \tag{2}$$

where α is a weighting factor, $\alpha \in [0,1]$ for class 1 and $1 - \alpha$ for another class; $(1 - p_i)^{\gamma}$ denotes the modulating factor; and γ denotes the tunable focusing parameter. The FL can balance the importance of positive and negative examples and differentiate between easy and hard examples by modulating the two factors α, and γ.

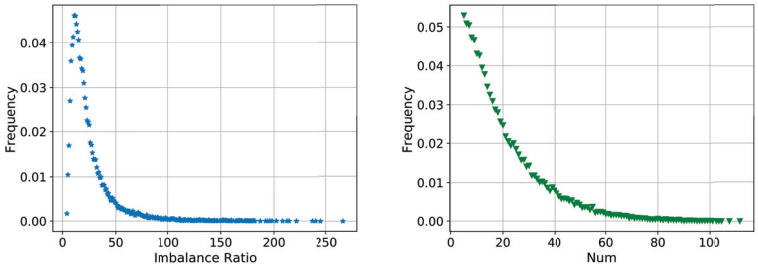
Inspired by the FL [44], we propose a novel crater edge segmentation loss function to optimize the proposed network. In contrast to FL, only the classification imbalance of data was considered when designing the loss function. In this paper, two data imbalance factors were considered, including the classification imbalance and the distribution imbalance of crater data, and the modulating factor of the loss function was set adaptively. We first calculated the imbalance characteristics of the crater data, which are shown in Figure 6. We used the data imbalance ratio (IR) to represent the crater data classification imbalance. This is the ratio between the numbers of majority class samples (background) and the minority class samples (object). The crater classification imbalance is shown in Figure 6a. Moreover, we counted the distribution imbalance ratio (DR) as the number of craters in each label's image, as shown in Figure 6b.

We set the parameters α' and γ' adaptively based on the DR and IR of the crater data for the CESL. The proposed craters edge segmentation loss function can be computed as

$$CESL(p_i) = -\alpha'(1 - p_i)^{\gamma'} log(p_i) \tag{3}$$

where α' is used to adjust the weights of different categories, and γ' is employed to differentiate between easy and hard examples. In this work, our goal was to accurately detect lunar craters. Some crater images are easy to distinguish, while others are difficult

to distinguish. Simple examples show that the distribution of some craters is sparse and complete with little overlap, making these craters easy to detect. These crater images are shown in Figure 3a,b. The hard example shows that, in the lunar crater image, the distribution of craters is dense with high overlap, and the shape is incomplete. These crater images are shown in Figure 3b,c. α' and γ' were calculated based on the average value in each trained batch. IR_b and DR_b are the classification imbalance ratio and the distribution imbalance ratio in the b-batch of network training.



(**a**) Crater Classification Imbalance (**b**) Crater Distribution Imbalance

Figure 6. Data distribution statistics of impact craters. We randomly generated 30,000 crater training images to show the imbalanced distribution. (**a**) is the crater image classification imbalance, the x-axis is the data imbalance ratio (IR) [45], and the y-axis denotes the frequency distribution of the IR. (**b**) shows the distribution imbalance ratio (DR) of the craters, the x-axis denotes the number of craters in each training image, and the y-axis represents the frequency distribution of the crater number in the DEM images.

In our crater data, we found that classification imbalance was common in the training data of each DEM image, and we calculated the max IR to be about 266 times and the average IR to be about 26 times, as shown in Figure 6a. In the crater training image, the densest crater image has 112 craters, and the average crater number is 20, as shown in Figure 6b. We defined the data imbalance degree in three cases based on the imbalance characteristics of craters, namely, low classification imbalance, median classification imbalance, and high classification imbalance. To balance the proportions of the data distribution, we calculated the ratio of three imbalance degree cases, which are more balanced when the ratio is about 3:2:1 in the crater training data, and the range of the corresponding IR is $IR > 40$, $20 < IR <= 40$, and $IR > 40$. The value of α' was set by the degree of imbalance, where IR_b was used to adjust the data imbalance with different weights. α' was set as

$$\alpha' = \begin{cases} 0.2, & IR_b < 20 \\ 0.3, & 20 < IR_b <= 40 \\ 0.4, & IR_b > 40 \end{cases} \tag{4}$$

Moreover, in general, highly overlapping, dense data may have a bad effect on crater classification. Thus, we also considered the craters' sparse distribution characteristics to improve the crater classification accuracy by setting the different values of γ'. The craters' sparse distribution characteristics DR were represented by the crater number in the DEM images. We defined DR_b by the crater number in the DEM images to set γ'. The parameter γ' is defined as

$$\gamma' = \begin{cases} 2, & DR_b < 20 \\ 1, & 20 < DR_b <= 100 \\ 1.5, & DR_b > 100. \end{cases} \tag{5}$$

2.5. Crater Extraction Algorithm

The crater image segmentation results were obtained by AFNet. The results included activated pixels corresponding to the locations of the crater rims. We were able to extract crater positions and sizes from the crater image segmentation results through the post-pipeline method with the crater extraction algorithm based on the template matching method. Most impact craters are circular on the lunar surface. The craters are detected by the ring feature in the extraction algorithm. However, for overlapping craters, traditional methods (such as Hough transform, Candy) [46] cannot detect rings in the segmentation results efficiently. We used the more efficient match template algorithm in scikit-image [47] (an image processing library implemented in Python programming language) to extract crater positions. This method was used in [36,37] for crater edge extraction.

The proposed CEA received the prediction map I of the crater segmentation network and output the crater evaluation results. The crater extraction pipeline process is as follows. First, a prediction result is filtered by the binary threshold β, described as

$$p_i = \begin{cases} 1, & p_i \geq \beta \\ 0, & p_i < \beta \end{cases} \tag{6}$$

where p_i is the pixel intensity. p_i is set to 1 when p_i is greater than β; otherwise, p_i is set to 0. Then, the match template algorithm is applied to match the crater over a radius range with a maximum radius r_{max} and minimum radius r_{min}. The match template threshold P_m is used to choose the high confidence target. Lastly, an evaluation of whether the crater is correctly identified is carried out.

We detected the minimum radius r_{min} of the craters as 5 km and the maximum radius r_{max} as 40 km from the network prediction result by the CEA. This algorithm iteratively slides generated rings through the target, and it calculates the match threshold at each (x, y, r) coordinate to eliminate false target results, where (x, y) is the centralization of the generated ring, and r is the radius. Any (x, y, r) ring with a match probability greater than P_m is classified by the coordinate and radius constraints to get the correct crater, expressed as

$$[(x_i - \tilde{x_j})^2 + (y_i - \tilde{y_j})^2] / min(r_i, \tilde{r_j})^2 < D_{x,y} \tag{7}$$

$$|r_i - \tilde{r_j}| / min(r_i, \tilde{r_j}) < D_r \tag{8}$$

where (x_i, y_i) is the position of the crater c_i extracted from the prediction image I, x_i, y_i are the latitude and longitude of I, respectively, and r_i is the radius of the crater c_i. For the ground-truth image $\tilde{I}$, $(\tilde{x_i}, \tilde{y_i})$ presents the position corresponding to the crater c_i, $\tilde{x_i}$ is the latitude of the crater, $\tilde{y_i}$ is the longitude of the crater, and the radius of crater c_i is $\tilde{r_i}$. $D_{x,y}$ is the error threshold of the longitude and latitude, and D_r is the radius error threshold. When the detection crater meets these limits, it is regarded as the correct crater; otherwise, it is considered a false crater.

The pseudo-code of the efficient lunar crater detection ELCD algorithm includes crater edge prediction by the semantic segmentation network AFNet and the post-pipeline method with CEA, as described in Algorithm 1.

The input of the network contains the test DEM data Y with a pixel size of 256×256 for the DEM image, the number of batch image processes $|Z(\widetilde{k})|$, the crater classification number N_{class}, the trained network model $\overline{M}$, and the ground-truth of the crater image $\widetilde{Y}$. The outputs are the position and size of the crater and the evaluation of the crater detection results. First, the batch data $Y(i)$ of crater images in test set Y are transferred to the trained model $\overline{M}$ by the AFNet to obtain the prediction results $pred_{dem}$ of the network. Then, the prediction feature map $pred_{dem}$ is processed by binary threshold processing β, using the match template threshold P_m to filter out matching craters. The correctly identified craters are evaluated by the error constraints shown in Equations (7) and (8), and the results of the evaluation are counted using statistical functions $Count()$. Finally, the position and size of

the crater *Pos* and the evaluation results *Det* of the correctly identified craters are obtained using the mean results for the test crater DEM data Y.

Algorithm 1: Efficient Lunar Crater Detection Algorithm

Input :
A set of DEM images, test dataset Y;
Each batch test data $Y(i)$ has $|Z(\widetilde{k})|$DEM images;
Each DEM image $y_i \in Y(i)$ has a size of 256 × 256 pixels;
The category of pixel segmentation is N_{class} ;
The trained network model $\overline{M}$;
The ground-truth in the test dataset $\widetilde{Y}$;
Output:
the information about the position and size *Pos*, and the crater detection results *Det*;
begin
  $\widehat{Z}$ = [];
  Load test dataset Y;
  $Pre-processing$ test dataset Y by normalization;
  model.eval();
  for *Each batch* $Y(j)$ *in* $Y(i)$ **do**
    for *Each image y in* $Y(j)$ **do**
      $orig_{dem} = y[0]$;
      $true_{dem} = y[1]$;
      model.cuda(), $orig_{dem}$.cuda(), $true_{dem}$.cuda();
      $pred_{dem} = \overline{M}$ ($orig_{dem}$);
      $\widehat{Z}$.add ($pred_{dem}$);
    end
  end
  Match = []; *// the results of the crater match template*
  Pos = []; *// the information about the crater's position and size*
  Det = []; *// the crater detection results*
  Match-template(); *// calculate template matching*
  Count(); *// statistical crater detection performance*
  for *Each DEM* z_k *in* $\widehat{Z}$ **do**
    if $N_{class} = 2$ *and* $z_k > \beta$ **then**
      $z_k \leftarrow 1.0$;
    else
      $z_k \leftarrow 0.0$;
    end
    if *Match-template*$(z_k) > P_m$ **then**
      Match.add ($longitude, latitude, radius$);
    end
    Choose the correctly identified craters by Equations (7) and (8);
    Output result $Ps, Dt \leftarrow Count\ (Match, \widetilde{y}_k)$, $\widetilde{y}_k \in \widetilde{Y}$;
    Pos.add(*Ps*);
    Det.add (*Dt*);
  end
  Pos = mean(*Pos*);
  Det = mean(*Det*);
end
Output: *Pos, Det*

2.6. Experiments

In this section, we describe the experiments conducted to verify the performance of the proposed algorithm. The experiments involved the experimental setup, experimental datasets, evaluation metrics, and comparison algorithms. The details are given below.

2.6.1. Experimental Setup

The experiment was performed on a single GPU (NVIDIA GeForce RTX 3060, 64GB RAM, 8 core CPU) with CUDA 11.0 and PyTorch 1.7.1. The CE-Adam [48] optimizer was used to improve the capability of the network model, and the learning rate was set to 1×10^{-4}. The number of iterations in the network was set to epoch 100, and the batch size was set to 32. We conducted crater detection experiments on the lunar DEM datasets, where the input DEM image was 256 × 256 pixels in size. The crater edge semantic segmentation

network AFNet and crater edge extract results were evaluated using relevant evaluation criteria, as detailed in Section 2.6.3.

2.6.2. Datasets

In our experiment, we used lunar DEM data from the Lunar Reconnaissance Orbiter (LRO) and the Kaguya merged digital elevation model. The resolution of the DEM was about 59 m/pixel [49], and it spanned 180° W to 180° E and 60° S to 60° N . The global DEM map was downsampled to 118 m/pixel with a size of 92,160 × 30,720 pixels. This was used to randomly generate crater images that were 256 × 256 pixels in size.

Two lunar crater catalogs were used for the ground truth. The first catalog was termed Head [10], where the size of the crater was larger than 20 km in diameter. The other catalog was taken from Povilaitis [9], and the crater diameter size was 5–20 km. We used the combined catalog, termed Head-LROC, to train our model in this paper. The total numbers of Head and Povilaitis craters were 5186, and 19,337, respectively. The different distributions and diameter sizes of craters based on the Head-LROC catalog are shown in Figure 7. We can see that around 51.5% of craters had a diameter of less than 10 km, which accounts for more than half of all data. Moreover, around 78.8% of craters had a radius of less than 20 km, representing about three-quarters of all crater data. Only 1.3% of craters had a radius of greater than 100 km.

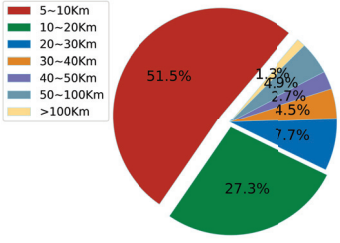


Figure 7. The distribution proportions of the different radius craters in the Head-LROC catalog [9,10].

In the experiment, the original crater images and ground-truth images were generated by the global DEM map and two lunar crater catalogs. The numbers of generated training sets, validation sets, and test sets were 30,000 DEM images, 3000, and 3000, respectively. The training set was processed by the random invert method. We randomly inverted θ to the DEM image using random number probability p, $p \in [0,1]$, where θ is defined as

$$\theta = \begin{cases} 0^\circ, & 0 <= p < 0.25 \\ 90^\circ, & 0.25 < p <= 0.5 \\ 180^\circ, & 0.50 < p <= 0.75 \\ 270^\circ, & 0.75 < p <= 1. \end{cases} \quad (9)$$

2.6.3. Evaluation Criteria

In two-stage crater detection algorithms, the performance of the prediction network may affect the final crater edge extraction result. When other parameters were fixed, the clearer the crater edge was segmented, the better the crater edge extraction result was. Thus, we first evaluated the performance of the proposed crater edge segmentation network, AFNet. The four metrics from common semantic segmentation criteria [23,24] were used to evaluate the proposed network model. We computed four metrics, the pixel accuracy (PA), mean pixel accuracy (MPA), mean intersection over union (MIoU), and frequency weighted intersection over union (FWIoU), to evaluate the performance of AFNet. Via an ablation study, we can prove the validity of our proposed model and the improved crater edge segmentation loss function.

After obtaining the crater image segmentation results, the crater positions and sizes can be obtained through the crater extraction algorithm. To evaluate the crater detection performance of the proposed ECLD algorithm, we used an evaluation method that is commonly used in machine learning to evaluate the precision (P), recall (R), and F_λ-score (F_1 or F_2) for each identified crater basis. The detection precision is the ratio of matching numbers N_{match} to detection numbers N_{detect} of craters. The recall was computed by the ratio of matching numbers N_{match} to the number of human-annotated N_{csv}, and the F_λ-score was used to balance the precision and recall. For the F_λ-score, λ denotes the tune parameter. When $\lambda > 1$, the recall is more important; otherwise, when $\lambda < 1$, the precision is more important for the model's evaluation. The detailed calculation process is described in [35,36].

Many truly existing craters were not marked in the ground truth; they were regarded as false negatives. In addition, in this paper, we used the combined lunar crater catalog Head-LROC [9,10]. The label of the training dataset was incomplete in the crater catalog, and some newly discovered craters were identified through network prediction. We calculated the discovery rate, that is, the false-positive rate for crater recognition. We used two methods to evaluate newly discovered craters. R^1_{new}, R^2_{new} was computed as

$$R^1_{new} = \frac{FP}{FP + TP} \tag{10}$$

$$R^2_{new} = \frac{FP}{FP + TP + FN} \tag{11}$$

where R^1_{new} denotes the ratio between the newly discovered craters and all recognized craters. TP denotes true positives and FP denotes false positives. The second evaluated method used was R^2_{new}, which shows the proportion of newly discovered craters to all impact craters, and FN indicates false negatives.

In the process of lunar crater recognition, the performance of the model was evaluated from the accuracy computation by the positions and sizes of the recognized craters. We calculated the latitude error (E^{lo}), longitude error (E^{la}), and radius error (E^r) to evaluate the network model using

$$E^{lo} = \frac{abs(lo^p - lo^t)}{2 \times (r^p + r^t)} \tag{12}$$

$$E^{la} = \frac{abs(la^p - la^t)}{2 \times (r^p + r^t)} \tag{13}$$

where lo^p denotes the predicted longitude value, and lo^t is the corresponding true longitude value of the crater. In Equation (13), la^p is the latitude value of the predicted crater, and the latitude value of the corresponding true crater is denoted as la^t.

The radius error (E^r) was calculated as follows:

$$E^r = \frac{abs(r^p - r^t)}{2 \times (r^p + r^t)} \tag{14}$$

where r^p denotes the radius of the predicted crater, and the corresponding true radius of the crater is indicated as r^t.

2.6.4. Compared Algorithms

The proposed algorithm ELCD was compared with five different crater detection algorithms using image segmentation technology that contained DeepMoon [35], ERU-Net [36], D-LinkNet [23], and SwiftNet [24]. The general procedure used for each algorithm was as follows:

- DeepMoon [35]: The basic idea of this algorithm is that deep learning based on the U-net network architecture is used to train the lunar crater DEM data to discover lunar craters.
- ERU-Net [36]: To improve the detection accuracy of lunar craters, ERU-Net introduced the residual network module to the U-Net network architecture to enhance the crater feature extraction ability.
- D-LinkNet [23]: D-LinkNet with high efficiency is often used for comparisons in crater detection. D-LinkNet is a semantic segmentation neural network that combines the encoder–decoder structure, dilated convolution, and a pre-trained encoder to carry out road extraction tasks.
- SwiftNet [24]: To verify the inference speed of the proposed model, we added SwiftNet to compare the network models. SwiftNet is a real-time semantic segmentation method based on residual network frameworks, which can achieve real-time detection for road-driving images.

3. Results

3.1. Ablation Study

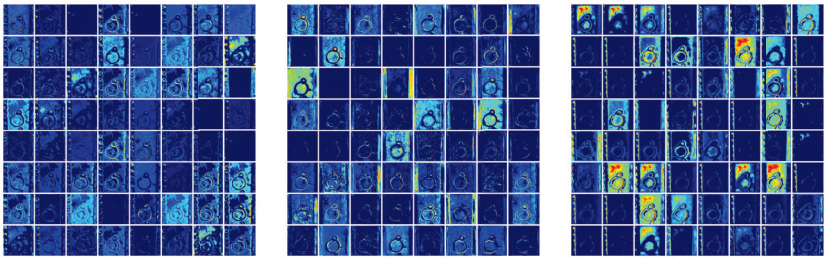
The ablation study on the AFNet explored the influences of different network structures and loss functions on the crater recognition accuracy. The proposed modules and three loss functions (LFs), CE, FL, and the proposed loss function CESL, were compared in the ablation study. The comparison network was initialized by using VGG-16 pre-training weights and normal initialization, where the ✓ denotes the use of the module, and VGG-16 denotes the basic network structure to give a better comparison. The results of the ablation study were obtained by evaluating PA, MPA, MIoU, and FWIoU in the crater validation data. The results are shown in Table 1, and the values in bold are the best values in each compared column.

In Table 1, we can see that the VGG-16-ECA increased by 0.1 and 0.2 MIoU in the CE and FL loss functions, and the MIoU increased by 0.1 and 0.2 MIoU compared with VGG-16 in VGG-16-MFF. When adding the attention machine module VGG-16-ECA and the efficient multiscale feature fusion module MFF, the MIoU obtained values of 73.0%, 74.4%, and 75.3% for the CE, FL, and CESL loss functions in AFNet. The AFNet network under the CESL achieved the best performance of 96.8%, 82.8%, 75.2%, and 94.3% for PA, MPA, MIoU, and FWIoU, respectively. The CESL considers crater data imbalance in classification and distributions and can balance the importance of positive and negative examples by adaptively setting the loss function weights. The proposed CESL loss function obtained a better performance in the compared network structures relative to CE and FL.

We also show several feature maps of a crater image sample at *decoder4* with the VGG-16, VGG-16-ECA, and AFNet network structures in Figure 8. We found that the output features had a clear distinction in AFNet and VGG-16-ECA compared with VGG-16. Some chance information was strengthened, while other chance information was weakened. AFNet and VGG-16-ECA included the attention mechanism ECA, which strengthens some important features to quickly distinguish the edges of craters from their backgrounds.

Table 1. Ablation experiment of the proposed modules on the DEM data.

Network Structures	LFs	ECA	MFF	PA (%)	MPA (%)	MIoU (%)	FWIoU (%)
VGG-16	CE			96.3%	80.0%	72.1%	93.5%
VGG-16-ECA	CE	✓		96.4%	80.6%	72.9%	93.7%
VGG-16-ECA	FL	✓		96.6%	81.6%	73.9%	94.0%
VGG-16-MFF	CE		✓	96.5%	81.1%	73.5%	93.9%
VGG-16-MFF	FL		✓	96.6%	81.8%	74.1%	94.0%
VGG-16-ECA-MFF (AFNet)	CE	✓	✓	96.5%	80.7%	73.0%	93.8%
VGG-16-ECA-MFF (AFNet)	FL	✓	✓	96.7%	82.0%	74.4%	94.1%
VGG-16-ECA-MFF (AFNet)	CESL	✓	✓	**96.8%**	**82.8%**	**75.2%**	**94.3%**

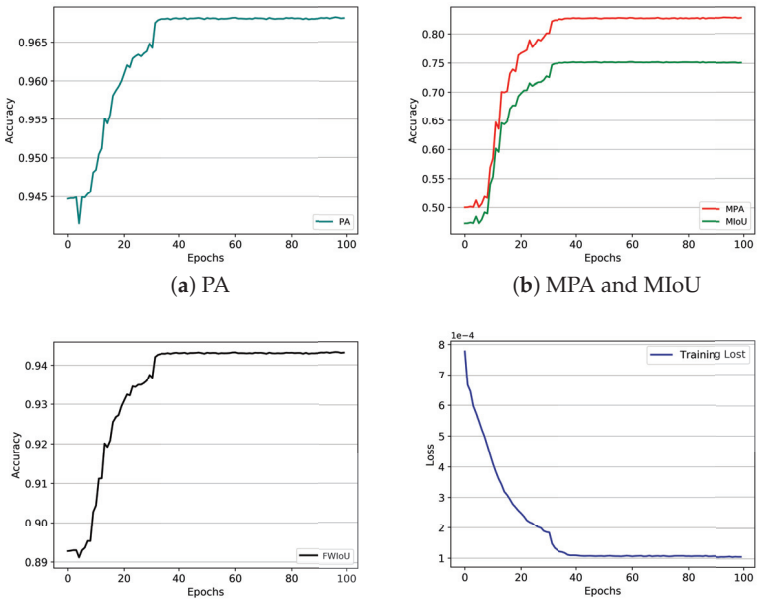


(**a**) VGG-16 (**b**) VGG-16-ECA (**c**) AFNet

Figure 8. Comparison of partial output results of different network structures in decoder4. (**a**) Partial output feature maps with the basic model VGG-16, (**b**) partial output feature maps with VGG-16-ECA, (**c**) partial output feature maps with AFNet.

3.2. The Evaluation Results for AFNet

In the iterative process of network training, the values of PA, MPA, MIoU, and FWIoU for AFNet in the validation set are shown in Figure 9. The accuracy of all evaluation criteria increased with the epoch. When the network was in about epoch 35 of network training, the network began to converge. The proposed model achieved a pixel accuracy of 96.8%, as shown in Figure 9a; the mean pixel accuracy was 82.8%, and the MIoU was 75.2%, as shown in Figure 9b. The FWIoU was 94.3%, as shown in Figure 9c. The training loss of the AFNet is shown in Figure 9d. We can see that the initial loss function was very small under the VGG-16 pre-training weight initialization, and the network had a faster convergence speed to allow it to obtain the best performance.



(**a**) PA (**b**) MPA and MIoU

(**c**) FWIoU (**d**) Training Loss

Figure 9. Semantic segmentation results on the validation set and training set of DEM data, (**a**–**c**) show the results of the validation set and c denotes the results of the training set. (**a**) is PA, (**b**) represents the MAP and MIoU, (**c**) denotes the FWIoU, and (**d**) is the training loss of the training set.

The network prediction results with AFNet are shown in Figure 10. The top figure denotes the ground truth of the DEM images, and the bottom figure shows the edge segmentation results. In lunar catalogs, some crater labeling is incomplete with small and shallow craters missing, and some obvious craters are not labeled, which may affect the crater detection accuracy. However, AFNet was used to recognize the crater edges through the classification of each pixel. We can see that the proposed AFNet network was able to segment crater edges with different characteristics.

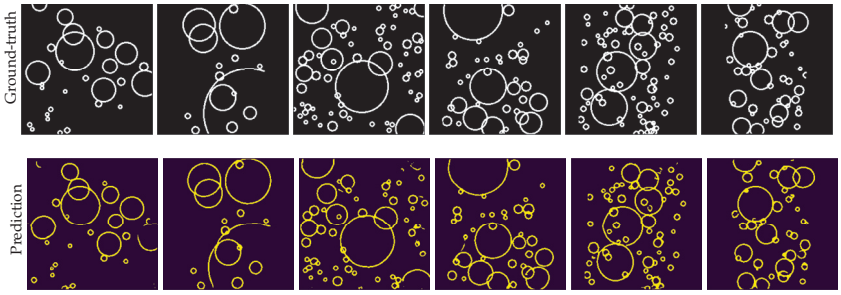


Figure 10. Crater edge segmentation prediction results based on the AFNet for the DEM data. The figure shows the ground truth of the DEM images, while the bottom figure denotes the crater edge segmentation prediction results.

3.3. The Evaluation Results for the ELCD

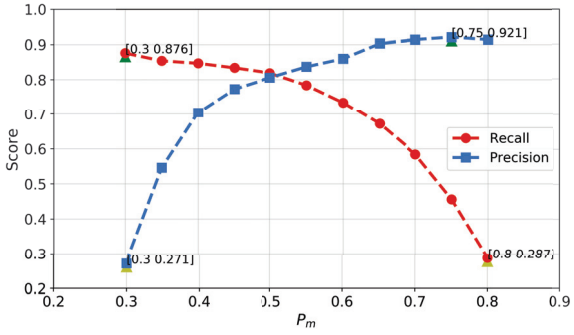
We evaluated the performance of the ELCD based on the edge segmentation network and crater extraction algorithm by detecting the crater radius, latitude, and longitude. Moreover, we computed the precision, recall, F_1, F_2, and the errors in the latitude, longitude, and radius of the crater for the match template method. In order to compare with other crater methods, we calculated the detection results of craters with a radius of 5–40 km. The error threshold of the longitude and latitude $D_{x,y}$ was set to 1.8, the radius error threshold D_r was set to 0.1, and the binary threshold β was set to 0.1. We tuned the match threshold P_m of the match template. For further details about the parameter setting process, refer to [35]. We evaluated the various metrics when the parameter of the match template threshold P_m ranged from 0.3 to 0.8 with an interval of 0.05. The average crater edge extraction resulted in different match threshold values P_m, as shown in Table 2. The best value in each compared row is presented in bold, and the gray column indicates the best tuning parameters.

In Table 2, we can see that the values of precision, F_1, and F_2 increased as P_m increased, while the values of recall and other metrics decreased as P_m increased. A high precision rate of 92.1% was obtained when P_m was 0.75 and the error values of E^{lo}, E^{la}, and E_r were also minimal. When γ was set to 0.3, the value of recall was maximal and more new craters were obtained under the maximum error values of E^{lo}, E^{la}, and E_r. New craters accounted for 41.9% and 70.2%, as shown by R^1_{new} and R^2_{new}. F_1 can balance the value of precision and recall. The best F_1 was 79.4% when P_m was set to 0.5, where the precision was 80.6%, the recall was 81.9%, and the error values of E^{lo}, E^{la}, and E^r were relatively small, at 12.0%, 9.8%, and 6.6%, respectively. F_2 pays more attention to the recall evaluation. When P_m was 0.45, F_2 obtained the best value of 80.9%. In this paper, in accordance with [35,36], we used F_1 and F_2 to evaluate the ELCD algorithm.

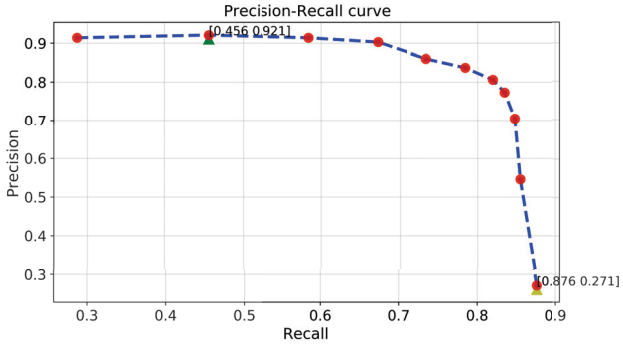
The precision and recall curves of the ELCD algorithm are shown in Figure 11, where the upper green triangle represents the maximal point, and the yellow triangle denotes the minimal value point. Figure 11a is the score of precision and recall with the different match thresholds P_m. The focus of these two lines is that P_m is about equal to 0.5, which is a balance point between precision and recall. The relation curve of the precision and recall curves is shown in Figure 11b.

Table 2. Crater edge extraction results of test sets in terms of various match thresholds P_m.

Metrics	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8
Precision	27.1%	54.6%	70.5%	77.3%	80.6%	83.7%	86.0%	90.3%	91.4%	**92.1%**	91.4%
Recall	**87.6%**	85.4%	84.7%	83.4%	81.9%	78.4%	73.4%	67.2%	58.4%	45.6%	28.7%
F_1	40.0%	64.7%	74.9%	78.3%	**79.4%**	79.0%	77.2%	74.9%	69.2%	58.5%	41.6%
F_2	58.4%	74.9%	80.0%	**80.9%**	80.6%	78.3%	74.6%	69.9%	62.0%	49.8%	32.6%
R^1_{new}	**41.9%**	29.9%	21.4%	17.1%	14.9%	12.7%	11.1%	8.1%	7.2%	6.6%	6.9%
R^2_{new}	**70.2%**	41.6%	26.4%	19.9%	16.7%	13.5%	11.0%	7.2%	5.5%	4.0%	2.9%
E^{lo}	17.5%	13.9%	12.7%	11.3%	12.0%	10.7%	9.3%	9.4%	8.8%	9.6%	**8.2%**
E^{la}	17.0%	13.7%	11.3%	10.6%	9.8%	9.2%	8.3%	7.7%	7.4%	7.0%	**6.8%**
E^r	13.0%	9.2%	8.0%	7.3%	6.6%	5.7%	4.8%	4.6%	4.2%	4.1%	**3.7%**



(**a**) P/R score with the P_m



(**b**) $P-R$ curve

Figure 11. Precision/recall curve for the crater detection results.

3.4. Comparison of Multiple Crater Detection Methods

In this section, we present an evaluation of the comparison results with ELCD under different crater detection methods using the test set. $P_m = 0.5$ is balance point between precision and recall. As shown in Figure 11a, we used the result where P_m was 0.5 as a comparison of ELCD. We also measured the computation complexity with different network architectures. In this paper, the billions of floating-point operations (FLOPs), network parameters (Params), and the number of processed frames per second (FPS) were used to evaluate the computational complexity of the trained networks. In the FPS computation, in accordance with [24], we set the test batch size as 1.

The average crater extraction results under various crater detection algorithms are shown in Table 3. In Table 3, we can see that the DeepMoon increased the recall and the proportion of newly discovered craters, and ERU-Net obtained a low detection error

for the crater radius, respectively. SwiftNet and D-linkNet had relatively poor detection accuracy levels, but they had the lowest FLOPs and network parameters. The crater detection algorithm required not only a high detection accuracy due to autonomous landing requirements for deep space probes in the deep space environment, but the crater detection algorithm should have a fast detection speed. The SwiftNet and D-linkNet network structures were designed for the real-time target detection of road-driving images. They have fewer parameters, low FLOPs, and high FPS during the running of the network to meet the needs of real-time detection. However, as the SwiftNet and D-linkNet network structures are simple network structures, they are inefficient for complex crater detection problems, and they perform poorly in lunar crater detection compared with other networks such as DeepMoon, ERU-Net, and the proposed algorithm. DeepMoon and ERU-Net achieved good crater detection results compared with the SwiftNet and D-linkNet network structures, but they require more computational resources, and the network computation speed of FPS is also lower.

Table 3. Comparison of the detection results of test sets under various crater detection algorithms.

Algorithms	P	R	F_1	F_2	R^1_{new}	R^2_{new}	E^{lo}	E^{la}	E^r	FLOPs (G)	Params (M)	FPS (HZ)
DeepMoon [35]	56.0%	**92.0%**	66.2%	72.9%	**40.0%**	**42.0%**	14.0%	11.0%	8.0%	74.3	10.28	8.7
ERU-Net [36]	75.4%	81.2%	78.1%	78.5%	18.3%	21.5%	**9.9%**	10.0%	7.8%	183.3	23.7	4.3
D-LinkNet [23]	77.2%	68.3%	61.2%	55.1%	17.3%	17.1%	10.1%	10.0%	7.3%	6.0	21.0	46.4
SwiftNet [24]	77.1%	52.6%	61.4%	56.1%	17.0%	13.3%	22.9%	19.9%	13.2%	**3.2**	**11.8**	60.2
ELCD (**our**)	**80.6%**	81.9%	**79.4%**	**80.6%**	14.9%	16.7%	12.0%	**9.8%**	**6.6%**	43.7	21.8	**73.2**

In Table 3, the proposed algorithm is shown to achieve better crater detection precision (P) and F_1, F_2 scores than the DeepMoon, SwiftNet, D-linkNet, and ERU-Net network structures with minimal E^{la} and E^r errors. Moreover, ELCD has a faster inference speed than the other algorithms. The proposed model combines the encoder, feature fusion, and decoder processes to achieve good network parallelism to speed up the network inference speed. The proposed ELCD has lower FLOPs than the DeepMoon and ERU-Net methods, and the total FLOPs in ELCD were shown to be about 1.7 times and 4.1 times lower than the values of DeepMoon and ERU-Net, respectively. For the FPS measure, although the parameters of the ELCD were not lower than those of DeepMoon and ERU-Net, the total FPS of the ELCD was about 8 times and 17 times higher than the values of DeepMoon and ERU-Net, respectively. Thus, the proposed ELCD algorithm achieved the best crater detection results with relatively few parameters and a low network complexity. It can achieve a balance between crater detection precision and network computation efficiency.

A comparison of the results obtained with different crater detection methods is shown in Figure 12. Each row represents the detection result of all compared crater methods for the same types of crater data. Each column represents the performance of the same detection method in different types of craters with varying degrees of classification and distribution imbalance. IR is the classification imbalance ratio, and DR denotes the distribution imbalance ratio, which was computed by the number of craters in each image label. The details are presented in Section 2.4. The greater the DR is, the denser the crater images are, and relatively speaking, the smaller the IR is. The original DEM image shown in Figure 12a,b is the ground truth, and Figure 12a–g denotes the compared algorithms. The blue circle denotes the correctly detected craters, the green circle is the newly detected craters, and the red circle is unrecognized craters. We can see that D-LinkNet and SwiftNet performed poorly for crater detection, especially for dense crater data. There are many incorrectly detected craters marked as red circles in Figure 12f,g. DeepMoon and ERU-Net could detect most of the labeled craters in contrast to D-LinkNet and SwiftNet, but they performed

poorly for large craters. For example, in IR = 9.3, DR = 49 and IR = 7.5, DR = 49, DeepMoon could not detect the large crater that is represented by the red circle in Figure 12d,e. In the third column, we can see that the proposed model increased the accuracy of crater detection compared with the other models for craters of different densities and sizes, as shown in Figure 12c. Moreover, the proposed model was able to detect some new unlabeled craters. However, small craters with a high degree of overlap in the DEM data were difficult to identify with high precision using DEM data for all compared algorithms. The proposed model regarded such craters as noise and could not detect them well.

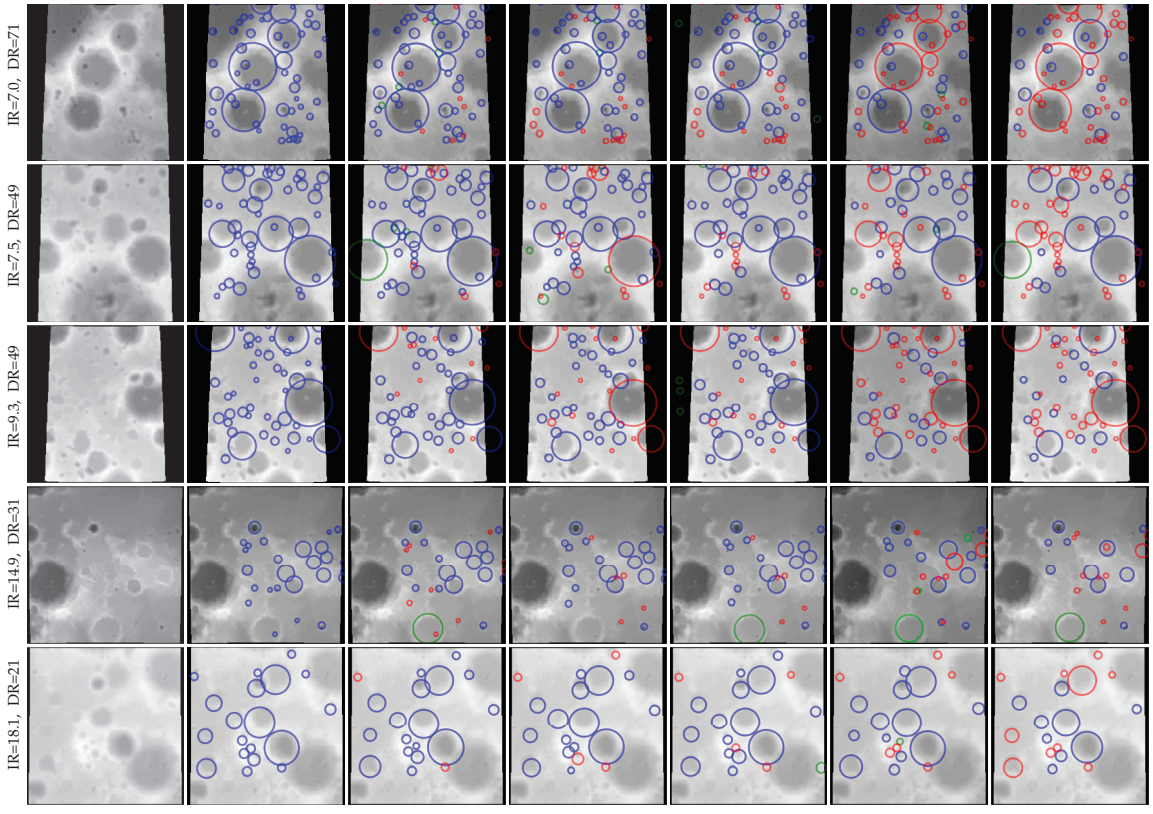


Figure 12. Comparison of the results of test sets using different crater detection methods for DEM data. (**a**) The original lunar DEM images in the test set. (**b**) The ground-truth DEM image. (**c**) The detection results obtained with DeepMoon based on U-net [35]. (**d**) The recognition results obtained with the ERU-Net network [36]. (**e**) The detection results obtained with D-LinkNet with the ResNet-18 network [23]. (**f**) The detection results obtained with SwiftNet (**g**), designed by the paper [24]. In the figure, the blue circles represent correctly recognized crates, the green circles denote new craters discovered by compared methods, and the red circle indicates unrecognized craters.

4. Discussion

With the application of deep learning techniques, great progress has been made in automated impact crater detection. The proposed method builds an efficient crater edge prediction network with a lightweight attention mechanism module and a multiscale feature fusion module to recognize crater edges from digital elevation models. The experimental results show that the presented method achieves high precision and recall rates and a

fast detection speed when undergoing lunar crater detection, mainly due to the following reasons: (1) we used the digital elevation model as the crater data, which contain abundant 3D morphology and topography morphological characteristics and are insensitive to light; (2) the proposed crater edge segmentation network is an efficient model to improve the accuracy of crater detection. The proposed network uses a lightweight attention mechanism module to enhance the feature extraction capability of the network encoder and designs a multiscale feature fusion module that fuses multi-level different resolution feature maps to reduce information loss in the network encoder; and (3) considering the imbalance of classification and different density distributions of craters, we proposed an efficient crater edge segmentation loss function to optimize the network performance.

In the experimental results, Table 1 shows that the multiscale feature fusion module can increase the crater detection accuracy, and it shows that the proposed crater loss function can achieve the best crater edge segmentation results. Figure 8 shows that the attention mechanism module can strengthen some chance information about craters and weaken other chance information, which can strengthen the importance of crater features to allow the edges of craters to be quickly distinguished from their backgrounds. Figure 9 shows that the CESL can improve the ability of the network to obtain optimal solutions and can speed up the convergence of the improved model. The final crater detection results show that the proposed model, which includes the attention mechanism module and the multiscale feature fusion module, can achieve more fine-grained segmentation for crater edges with different characteristics, as shown in Figure 10. In Table 3 and Figure 12, which shows a comparison of the different crater detection methods, the proposed model is shown to achieve the best detection performance with minimal errors in E^{la} and E^{r}. Compared with other real-time target detection methods, this method has a faster reasoning speed. Compared with the survey of the global lunar orbiter laser altimeter (LOLA) dataset of the Moon, the algorithm can detect the marked craters on the lunar surface more accurately and can detect some undiscovered craters. There are some false and ambiguous markers in the global LOLA dataset, and the proposed algorithm can correct false positives in the original data. Moreover, the newly discovered craters can increase the size of the original data set.

The discovery of impact craters is important for studying the evolution of the Moon. There are many small craters on the Moon's surface, and they influence the estimation of the Moon's age. However, the study still has some limitations with regard to small crater detection. Most crater digital elevation models have a lower resolution than the optical image and other higher-resolution images. Some craters that are too small appear as points in DEM images, and they are likely to be ignored or considered to be noise and thus cannot be detected successfully using a digital elevation model. The optical image has a high resolution, but it is sensitive to illumination. Thus, determining how to avoid the impact of light on impact craters in optical images or fusing the optical image and the digital elevation model to improve the small crater detection accuracy deserve further attention in the future.

5. Conclusions

In this paper, an efficient lunar crater detection algorithm, AFNet, based on the segmentation convolutional neural network was proposed to improve the crater detection accuracy and speed. Based on the VGG-16 network architecture, a lightweight attention mechanism module was introduced to enhance the extraction of important crater features in the network encoder. The proposed model uses a new feature fusion method that fuses multi-level different feature maps obtained from the network encoder to reduce the information loss of the output map in the network decoder. Then, considering the classification and distribution imbalance of the crater data, the crater edge segmentation loss function was used to improve the optimization performance of the proposed model. Last, the crater positions were extracted by the crater edge extract algorithm based on the match template method. The proposed model was applied to two crater catalogs and

compared with four state-of-the-art crater detection algorithms. The results demonstrate that the ELCD achieved an inference speed of about 73 HZ and a precision of 80.6% for lunar crater detection in a DEM image with 256 × 256 pixels on *GeForce RTX 3060*, and it obtained the best accuracy of 79.4% for F_1 and 80.6% for F_2 compared with the other crater detection models. Moreover, the ELCD can be used to discover new craters and expand the size of the original data set. It is hoped that this algorithm will further improve the accuracy of lunar age estimation and the positioning accuracy of spacecraft landing. For future work, the network structure should be further optimized so that the model can improve its real-time detection speed and achieve a high crater detection accuracy in the detection of impact craters of different sizes.

Author Contributions: Conceptualization, L.F. and K.Z.; formal analysis, L.F. and X.W.; funding acquisition, J.Y.; investigation, J.Y.; methodology, L.F., J.Y. and K.Z.; project administration, J.Y.; software, L.F. and K.Z.; supervision, J.Y.; visualization, L.F. and X.W.; writing—original draft, L.F.; writing—review and editing, L.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (Grant No. 62076127).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank R. Povilaitis and J. Head for providing the 5–20 km and >20 km dataset of lunar craters. In addition, the authors would like to thank the reviewers for their valuable comments and suggestions.

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

Abbreviations

The following abbreviations are used in this manuscript:

ELCD	Efficient lunar crater detection
DEM	Digital elevation model
CNNs	Convolution neural networks
SAR	Synthetic aperture radar
AFNet	Attention mechanisms and multiscale feature fusion networks
CESL	Crater edge segmentation loss
CEA	Crater extraction algorithm
MFF	Multiscale feature fusion
VGG-16	Visual geometry group-16
ECA	Efficient channel attention
IR	Data imbalance ratio
DR	Distribution imbalance ratio
CE	Cross-entropy
FL	Focal loss
LRO	Lunar reconnaissance orbiter
PA	Pixel accuracy
MPA	Mean pixel accuracy
MIoU	Mean intersection over union
FWIoU	Frequency weighted intersection over union
FLOPs	Floating-point operations
FPS	Frames per second
P	Precision
R	Recall

References

1. Strom, R.; Malhotra, R.; Ito, T.; Yoshida, F.; Kring, D. Origin of Impacting Objects in the Inner Solar System. In Proceedings of the AGU Fall Meeting Abstracts, San Francisco, CA, USA, 5–9 December 2005; Volume 2005, p. P42A-01.
2. Schmidt, M.W.; Kraettli, G. Experimental Crystallization of the Lunar Magma Ocean, Initial Selenotherm and Density Stratification, and Implications for Crust Formation, Overturn and the Bulk Silicate Moon Composition. *J. Geophys. Res. (Planets)* **2022**, *127*, e07187. [CrossRef]
3. Bottke, W.F.; Norman, M.D. The late heavy bombardment. *Annu. Rev. Earth Planet. Sci.* **2017**, *45*, 619–647. [CrossRef]
4. Kereszturi, A.; Steinmann, V. Terra-mare comparison of small young craters on the Moon. *Icarus* **2019**, *322*, 54–68. [CrossRef]
5. Xu, L.; Qiao, L.; Xie, M.; Wu, Y. Formation age of lunar Lalande crater and its implications for the source region of the KREEP-rich meteorite Sayh al Uhaymir 169. *Icarus* **2022**, *386*, 115166. [CrossRef]
6. Martin, I.; Parkes, S.; Dunstan, M. Modeling cratered surfaces with real and synthetic terrain for testing planetary landers. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 2916–2928. [CrossRef]
7. Song, J.; Rondao, D.; Aouf, N. Deep learning-based spacecraft relative navigation methods: A survey. *Acta Astronaut.* **2022**, *191*, 22–40. [CrossRef]
8. Kereszturi, A.; Tomka, R.; Steinmann, V. Testing statistical impact crater analysis in permanently shadowed lunar polar regions. *Icarus* **2022**, *376*, 114879. [CrossRef]
9. Povilaitis, R.; Robinson, M.; Van der Bogert, C.; Hiesinger, H.; Meyer, H.; Ostrach, L. Crater density differences: Exploring regional resurfacing, secondary crater populations, and crater saturation equilibrium on the moon. *Planet. Space Sci.* **2018**, *162*, 41–51. [CrossRef]
10. Head III, J.W.; Fassett, C.I.; Kadish, S.J.; Smith, D.E.; Zuber, M.T.; Neumann, G.A.; Mazarico, E. Global distribution of large lunar craters: Implications for resurfacing and impactor populations. *Science* **2010**, *329*, 1504–1507. [CrossRef]
11. Troglio, G.; Le Moigne, J.; Benediktsson, J.A.; Moser, G.; Serpico, S.B. Automatic extraction of ellipsoidal features for planetary image registration. *IEEE Geosci. Remote Sens. Lett.* **2011**, *9*, 95–99. [CrossRef]
12. Galloway, M.J.; Benedix, G.K.; Bland, P.A.; Paxman, J.; Towner, M.C.; Tan, T. Automated crater detection and counting using the Hough transform. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 1579–1583.
13. Di, K.; Li, W.; Yue, Z.; Sun, Y.; Liu, Y. A machine learning approach to crater detection from topographic data. *Adv. Space Res.* **2014**, *54*, 2419–2429. [CrossRef]
14. Bandeira, L.; Saraiva, J.; Pina, P. Impact crater recognition on Mars based on a probability volume created by template matching. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 4008–4015. [CrossRef]
15. Emami, E.; Bebis, G.; Nefian, A.; Fong, T. Automatic crater detection using convex grouping and convolutional neural networks. In Proceedings of the International Symposium on Visual Computing, Las Vegas, NV, USA, 14–16 December 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 213–224.
16. He, J.; Cui, H.; Feng, J. Edge information based crater detection and matching for lunar exploration. In Proceedings of the 2010 International Conference on Intelligent Control and Information Processing, Dalian, China, 13–15 August 2010; IEEE: Hoboken, NJ, USA, 2010; pp. 302–307.
17. Chen, M.; Liu, D.; Qian, K.; Li, J.; Lei, M.; Zhou, Y. Lunar crater detection based on terrain analysis and mathematical morphology methods using digital elevation models. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 3681–3692. [CrossRef]
18. Barata, T.; Alves, E.I.; Saraiva, J.; Pina, P. Automatic recognition of impact craters on the surface of Mars. In Proceedings of the International Conference Image Analysis and Recognition, Porto, Portugal, 29 September–1 October 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 489–496.
19. Urbach, E.R.; Stepinski, T.F. Automatic detection of sub-km craters in high resolution planetary images. *Planet. Space Sci.* **2009**, *57*, 880–887. [CrossRef]
20. Salamunićcar, G.; Lončarić, S. Application of machine learning using support vector machines for crater detection from Martian digital topography data. *38th COSPAR Sci. Assem.* **2010**, *38*, 3.
21. Kang, Z.; Wang, X.; Hu, T.; Yang, J. Coarse-to-fine extraction of small-scale lunar impact craters from the CCD images of the Chang'E lunar orbiters. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 181–193. [CrossRef]
22. Wang, Y.; Wu, B. Active machine learning approach for crater detection from planetary imagery and digital elevation models. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5777–5789. [CrossRef]
23. Zhou, L.; Zhang, C.; Wu, M. D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 182–186.
24. Orsic, M.; Kreso, I.; Bevandic, P.; Segvic, S. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 12607–12616.
25. Li, Y.; Du, L.; Wei, D. Multiscale CNN based on component analysis for SAR ATR. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–12. [CrossRef]
26. Liao, L.; Du, L.; Guo, Y. Semi-supervised SAR target detection based on an improved faster R-CNN. *Remote Sens.* **2021**, *14*, 143. [CrossRef]

27. DeLatte, D.; Crites, S.T.; Guttenberg, N.; Yairi, T. Automated crater detection algorithms from a machine learning perspective in the convolutional neural network era. *Adv. Space Res.* **2019**, *64*, 1615–1628. [CrossRef]
28. Downes, L.M.; Steiner, T.J.; How, J.P. Neural Network Approach to Crater Detection for Lunar Terrain Relative Navigation. *J. Aerosp. Inf. Syst.* **2021**, *18*, 391–403. [CrossRef]
29. Silvestrini, S.; Piccinin, M.; Zanotti, G.; Brandonisio, A.; Bloise, I.; Feruglio, L.; Lunghi, P.; Lavagna, M.; Varile, M. Optical navigation for Lunar landing based on Convolutional Neural Network crater detector. *Aerosp. Sci. Technol.* **2022**, *123*, 107503. [CrossRef]
30. Zhang, Z.; Liu, Q.; Wang, Y. Road extraction by deep residual u-net. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 749–753. [CrossRef]
31. Li, W.; Zhou, B.; Hsu, C.Y.; Li, Y.; Ren, F. Recognizing terrain features on terrestrial surface using a deep learning model: An example with crater detection. In Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery, Los Angeles, CA, USA, 7 November 2017; pp. 33–36.
32. Tewari, A.; Verma, V.; Srivastava, P.; Jain, V.; Khanna, N. Automated crater detection from co-registered optical images, elevation maps and slope maps using deep learning. *Planet. Space Sci.* **2022**, *218*, 105500. [CrossRef]
33. Yang, S.; Cai, Z. High-Resolution Feature Pyramid Network for Automatic Crater Detection on Mars. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–12. [CrossRef]
34. Wang, H.; Jiang, J.; Zhang, G. CraterIDNet: An end-to-end fully convolutional neural network for crater detection and identification in remotely sensed planetary images. *Remote Sens.* **2018**, *10*, 1067. [CrossRef]
35. Silburt, A.; Ali-Dib, M.; Zhu, C.; Jackson, A.; Valencia, D.; Kissin, Y.; Tamayo, D.; Menou, K. Lunar crater identification via deep learning. *Icarus* **2019**, *317*, 27–38. [CrossRef]
36. Wang, S.; Fan, Z.; Li, Z.; Zhang, H.; Wei, C. An effective lunar crater recognition algorithm based on convolutional neural network. *Remote Sens.* **2020**, *12*, 2694. [CrossRef]
37. DeLatte, D.M.; Crites, S.T.; Guttenberg, N.; Tasker, E.J.; Yairi, T. Segmentation convolutional neural networks for automatic crater detection on mars. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 2944–2957. [CrossRef]
38. 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; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
39. Lee, C.; Hogan, J. Automated crater detection with human level performance. *Comput. Geosci.* **2021**, *147*, 104645. [CrossRef]
40. Hu, Z.; Shi, T.; Cen, M.; Wang, J.; Zhao, X.; Zeng, C.; Zhou, Y.; Fan, Y.; Liu, Y.; Zhao, Z. Research progress on lunar and Martian concrete. *Constr. Build. Mater.* **2022**, *343*, 128117. [CrossRef]
41. Ketkar, N.; Moolayil, J. Introduction to pytorch. In *Deep Learning with Python*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 27–91.
42. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
43. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539.
44. 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. 2980–2988.
45. Van Hulse, J.; Khoshgoftaar, T.M.; Napolitano, A. Experimental perspectives on learning from imbalanced data. In Proceedings of the 24th International Conference on Machine Learning, Corvalis, OR, USA, 20–24 June 2007; pp. 935–942.
46. Bue, B.D.; Stepinski, T.F. Machine detection of Martian impact craters from digital topography data. *IEEE Trans. Geosci. Remote Sens.* **2006**, *45*, 265–274. [CrossRef]
47. 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]
48. Yong, H.; Huang, J.; Hua, X.; Zhang, L. Gradient centralization: A new optimization technique for deep neural networks. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 635–652.
49. Barker, M.; Mazarico, E.; Neumann, G.; Zuber, M.; Haruyama, J.; Smith, D. A new lunar digital elevation model from the Lunar Orbiter Laser Altimeter and SELENE Terrain Camera. *Icarus* **2016**, *273*, 346–355. [CrossRef]

remote sensing

MDPI

Article

Pomelo Tree Detection Method Based on Attention Mechanism and Cross-Layer Feature Fusion

Haotian Yuan [1,†], Kekun Huang [2,3,†], Chuanxian Ren [4], Yongzhu Xiong [3,5], Jieli Duan [1] and Zhou Yang [1,3,*]

[1] School of Engineering, South China Agricultural University, Guangzhou 510642, China
[2] School of Mathematics, Jiaying University, Meizhou 514015, China
[3] Guangdong Provincial Key Laboratory of Conservation and Precision Utilization of Characteristic Agricultural Resources in Mountainous Areas, Jiaying University, Meizhou 514015, China
[4] School of Mathematics, Sun Yat-sen University, Guangzhou 510275, China
[5] School of Geography and Tourism, Jiaying University, Meizhou 514015, China
* Correspondence: yangzhou@scau.edu.cn
† These authors contributed equally to this work.

Abstract: Deep learning is the subject of increasing research for fruit tree detection. Previously developed deep-learning-based models are either too large to perform real-time tasks or too small to extract good enough features. Moreover, there has been scarce research on the detection of pomelo trees. This paper proposes a pomelo tree-detection method that introduces the attention mechanism and a Ghost module into the lightweight model network, as well as a feature-fusion module to improve the feature-extraction ability and reduce computation. The proposed method was experimentally validated and showed better detection performance and fewer parameters than some state-of-the-art target-detection algorithms. The results indicate that our method is more suitable for pomelo tree detection.

Keywords: convolutional neural network; object detection; attention mechanism; remote-sensing image; pomelo tree detection

Citation: Yuan, H.; Huang, K.; Ren, C.; Xiong, Y.; Duan, J.; Yang, Z. Pomelo Tree Detection Method Based on Attention Mechanism and Cross-Layer Feature Fusion. *Remote Sens.* **2022**, *14*, 3902. https://doi.org/10.3390/rs14163902

Academic Editors: Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Received: 9 July 2022
Accepted: 9 August 2022
Published: 11 August 2022

1. Introduction

Citrus is the world's largest fruit group, and pomelo is the largest citrus fruit [1]. High-quality pectin can be extracted from the peel of the pomelo, and the pulp can be processed into juice and wine. Pomelo tree-planting information is vital to growers, as it can provide a basis to scientifically manage planting and improve income per unit area and unit time. Detecting the location and quantity of pomelo trees helps growers to develop precision and intelligence in orchard management, such as in fertilization and irrigation [2], pruning [3], and pesticide application [4], to reduce production costs, reduce environmental pollution, and improve fruit yield and quality [5]. However, in actual production, data are obtained largely manually, which requires much labor, and samples limited numbers of trees. This will result in inaccurate data analysis and unreliable experimental results. Therefore, a quick, non-destructive, and accurate pomelo tree-detection method is needed to replace manual inspection.

Remote sensing technology has developed rapidly, and studies have demonstrated its applicability to agriculture. For example, a plant water stress model was established by combining satellite remote-sensing data with ground agrometeorological data [6], an orange tree-detection model was established by a remote sensing platform combined with unmanned aerial vehicles (UAVs) and sensors [4], and phytophthora root rot (PRR) disease on avocado tree roots was detected by remote sensing and hyperspectral imaging [7]. Aerial and satellite remote sensing are limited by weather conditions and monitoring costs [8]. Compared with satellites, UAVs are less dependent on weather conditions [9], and they can be deployed in harsh environments with fast data collection [10].

Tree-detection methods for remote-sensing images are mainly based on three kinds of methods: traditional image processing, traditional machine learning, and deep learning. Traditional image processing-based tree-detection methods have no parameter-learning process, such as the local maximum method [11], watershed segmentation algorithm [12], and multi-scale segmentation algorithm [13]. Srestasathiern et al. [14] proposed a method for oil palm identification based on algorithms such as feature selection, semi-variance function calculation, and local maximum filtering. Dos Santos et al. [15] proposed a palm tree-detection method based on shadow extraction and template matching, which correctly detected 75.45% of the trees in a study area of about 95 square kilometers. However, traditional image-processing methods have lower recognition accuracy. Furthermore, they require manual setting of many parameters.

Traditional machine learning-based methods typically comprise steps such as feature extraction, image segmentation, classifier training, and prediction [16–18]. López-López et al. [19] proposed a method for detecting unhealthy trees based on image segmentation and support vector machine classifiers. Nevalainen et al. [20] proposed a method for tree detection and species classification, which includes tree detection using local maximum filtering, feature extraction, and tree species classification using random forest and artificial neural network methods. Wang et al. [21] proposed utilizing a gradient histogram operator and a support vector machine classifier to identify oil palm trees in UAV imagery. In general, traditional machine-learning-based methods outperform traditional image-processing-based methods. However, their feature extraction capability is insufficient, which limits them regarding achieving higher detection accuracy.

A deep-learning-based algorithm can extract complex structural information from huge amounts of high-dimensional data, using a neural network with multiple hidden layers to automatically learn features from the original image [22]. Convolutional Neural Networks (CNNs) are among the best-known deep learning-based methods owing to their good image interpretability. CNNs are widely used to solve agricultural production problems, including plant pest detection and classification [23,24], plant leaf identification and classification [25,26], weed identification and classification [27,28], fruit and vegetable harvesting and identification [29,30], and land cover classification [31,32].

Deep-learning-based algorithms have improved tree-detection performance. Li et al. [33] proposed a CNN-based framework for detection and counting of oil palm trees in high-resolution remote-sensing images, and this framework showed greater accuracy than three other models. Pibre et al. [34] proposed a tree-identification method using multi-scale sliding windows and neural networks. Wu et al. [35] researched the dead branches of apple trees in winter, using remote-sensing data collected by UAVs, and used Faster R-CNN to determine the number and location of trees. Zheng et al. [36] proposed a multi-type method to accurately detect oil palm trees and monitor their growth. The method is based on Faster R-CNN [37], and uses a refine pyramid feature module for feature extraction, which can integrate deep and shallow features to help distinguish similar classes and detect smaller oil palms. Osco et al. [38] proposed a method to estimate the number and location of citrus trees in an orchard using an estimated density map, and this method achieved higher F1-scores than Faster R-CNN and RetinaNet. Zheng et al. [39] proposed a coconut tree crown-detection method, which contains three major procedures: feature extraction, a multi-level Region Proposal Network (RPN), and a large-scale coconut tree-detection workflow. The method achieves a higher average F1-score than pure Faster R-CNN. Some methods based on domain adaption methods were proposed for tree detections [40,41]. They divide the data into a target domain that has few or no labels and a source domain that has many labels. These techniques achieve detection by applying the information acquired in the source domain to the target domain.

Most of the above methods use a two-stage target-detection network: (1) generation of candidate region proposals through a RPN; and (2) classification and bounding-box regression tasks for selected candidates. Two-stage detection networks more time-consuming and have lower computational efficiency than single-stage detection networks [42]. As a

single-stage target detector, SSD [43] and YOLO [44–46] treat target detection as a regression problem. YOLOx-nano [47], the lightweight version of the model, has fewer model parameters and runs faster, so it is suitable for real-time tasks, but it is not good enough due to the complex and changeable environment of remote-sensing systems in agriculture. Moreover, there is scant research on the detection of pomelo trees.

To improve recognition accuracy in complex environments, some researchers incorporated attention mechanisms into neural network models [48–50], and some researchers used feature-fusion modules that combine features at different scales to enable the network to extract richer features [51–53]. However, they did not consider that shallow and deep feature information in deep networks have complimentary characteristics.

In summary, the existing methods for tree detection have the following problems:

- The two-stage algorithm has good performance in tree detection, but the algorithm is complex, leading to computational inefficiency and slow detection.
- The one-stage algorithm runs faster than the two-stage one, but the model size is still too large for real-time application. The lightweight version of the one-stage algorithm is fast enough, but the feature-extraction ability is limited.
- Some studies used an attention mechanism and a feature-fusion module to improve feature-extraction ability. However, they did not consider the advantage of the complimentary characteristics between different layers.

To address the above problems, we propose a pomelo tree-detection method for UAV remote-sensing images based on YOLOx-nano; it utilizes the complementary characteristics of features at different levels to hierarchically aggregate rich information, thereby achieving more accurate detection and counting of pomelo trees. A hybrid attention mechanism module learns more representative features from the underlying features extracted from the backbone feature extraction network. A feature-fusion module utilizes the complementary characteristics of the extracted low-level detail information and high-level semantic information to perform cross-layer fusion of feature maps. A Ghost module replaces the convolution module for better computational efficiency.

In this study, we make the following contributions:

- A hybrid attention mechanism module weights the pixels of the feature map with channel attention and spatial attention to improve feature extraction and highlight pomelo tree regions in backgrounds;
- A feature-fusion module fuses the feature maps of different layers without greatly increasing computation, so it effectively aggregates feature maps;
- A Ghost module replaces the convolution module, reducing the number of parameters and the computational complexity of the deep network, so as to further improve the model-detection effect.

2. Materials and Methods

2.1. Materials

2.1.1. Image Data Collection

The remote-sensing dataset used in this study was obtained from an orchard of pomelo trees in ShiShan and Yanyang Town, Meizhou City, Guangdong Province, China (23°23′~24°56′N and 115°18′~116°56′E). Known as the "hometown of the pomelo", Meizhou is in the eastern part of Guangdong Province. The warm and humid climate, abundant rainfall, and deep and well-drained soil provide good conditions for pomelo cultivation. The test site covers an area of about 50 hectares, and the spacing of pomelo trees is 4 × 4 m. UAV remote-sensing images were collected from a quadrotor UAV (Phantom 4, DJi, Guangdong, China) equipped with a visual spectral (RGB) camera with a spatial resolution of 0.05 m. A total of 1222 UAV remote-sensing images (5642 × 3648 pixels) were collected, from an altitude of 120 m, with an overlap rate of 60%, in two areas heavily planted with pomelo trees, including images of other trees, houses, and roads, as shown in Figure 1.

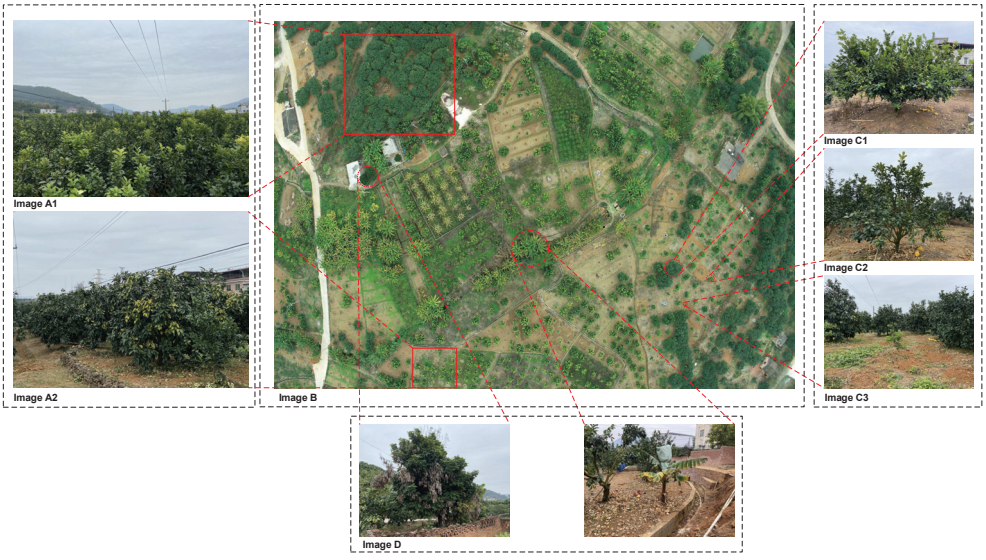


Figure 1. Aerial image of pomelo orchard. Image B is the original image from the vertical overhead shot of the UAV. Images A1 and A2 contain dense and sparse pomelos, respectively; images C1–C3 contain pomelo trees in the adult, middle-aged, and young-growth stages, respectively; image D contains other trees in the orchard.

The first dataset was collected at 10:00 on 26 December 2021 in Shishan town (24°26′N and 116°05′N), Meizhou City, Guangdong Province, China. The shooting location was on a flat terrain, as shown in Figure 2a–c. A large number of pomelo trees were planted in this region. We used UAVs for vertical overhead photography and collected 1022 UAV remote-sensing photos in this area.

The second dataset was captured at 16:00 on 16 January 2022 in Yanyang Town (24°22′N and 116°22′N), Meizhou City, Guangdong Province, China. The shooting area is located in hilly mountainous terrain. This dataset is more challenging than the first one. Due to the diverse topography undulations, pomelo trees were planted at various heights and were exposed to varying amounts of sunshine, as shown in Figure 2d–f. In addition, there is a large amount of other vegetation planted in mountainous areas with a complex environment of overgrown trees. For vertical overhead photography, we obtained 233 remote-sensing images using UAVs.

2.1.2. Image Annotation and Data Generation

We selected 20 and 8 images (5642 × 3648 pixels) from the Shishan and Yanyang Town UAV image database, which were cropped without overlap to obtain 1674 cropped images, respectively. Each image contained 0–20 pomelo trees, and was of the size 640 × 640 pixels, as shown in Figure 2. We used the open-source image-editing tool LabelImg to manually label the images, with one box to label a pomelo tree. For each dataset, we randomly selected 60% for training, 20% for validation, and 20% for testing. Half of the images in the training dataset were brightened, darkened, flipped, and scaled to increase the richness of the training sample data.

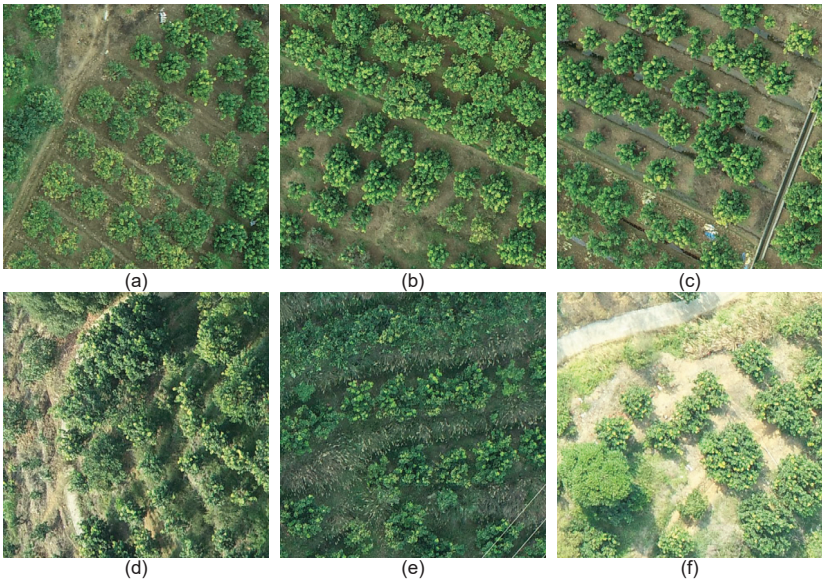


Figure 2. Samples of dataset 1 (**a**–**c**) and dataset 2 (**d**–**f**). The pomelo trees in dataset 1 are planted on a plain, under uniform lighting. The pomelo trees in dataset 2 were planted on hillsides with uneven distribution, under drastic lighting variation.

2.2. Proposed Method

We introduce an attention mechanism behind the backbone feature extraction network, which allows the model to focus on the pomelo canopy, improving the differentiation of the pomelo trees from the backdrop. A cross-layer feature-fusion model (CLFF) helps the network to more effectively fuse features at different layers so as to enrich feature information extracted by the model and improve model-detection capability. The CNN module can create a large number of similar feature maps, and it has been shown that these redundant feature maps enable the CNN's excellent feature-extraction capabilities [54]. The proposed method can reduce model parameters and improve feature extraction. The proposed network structure is shown in Figure 3.

2.2.1. Hybrid Attention Mechanism Module

Despite the increasing resolution of remote-sensing images, there is still ambiguity in the boundaries between objects, which increases false detection. In addition, the sizes of pomelo trees vary by growth stages, causing poor performance. To address this problem, we use an attention mechanism [55] to enhance the importance of target pixels in both channels and space, which can strengthen the information of pomelo trees and weaken background information by weighting the features extracted by the backbone feature network. The attention score indicates the degree of correlation between pixels and targets [56], and can be used to focus on pomelo trees and reduce the impact of canopy sizes.

The hybrid attention mechanism (Figure 4) has channel and spatial attention mechanisms, focusing on both channel and pixel point weighting during model training.

In the channel attention mechanism, maximum and average pooling are applied to the $h \times w \times c$ feature map to obtain two $1 \times w \times c$ feature strips, which are fed into the shared full-connected module, which contains two full-connected layers. The number of neurons in the first full connection is small, and the number in the second full connection is equal to the number of input channels. The resulting two features are summed, and the weight coefficients of each $1 \times w \times c$ channel are obtained by the sigmoid function. The weight coefficients are multiplied with the input feature map.

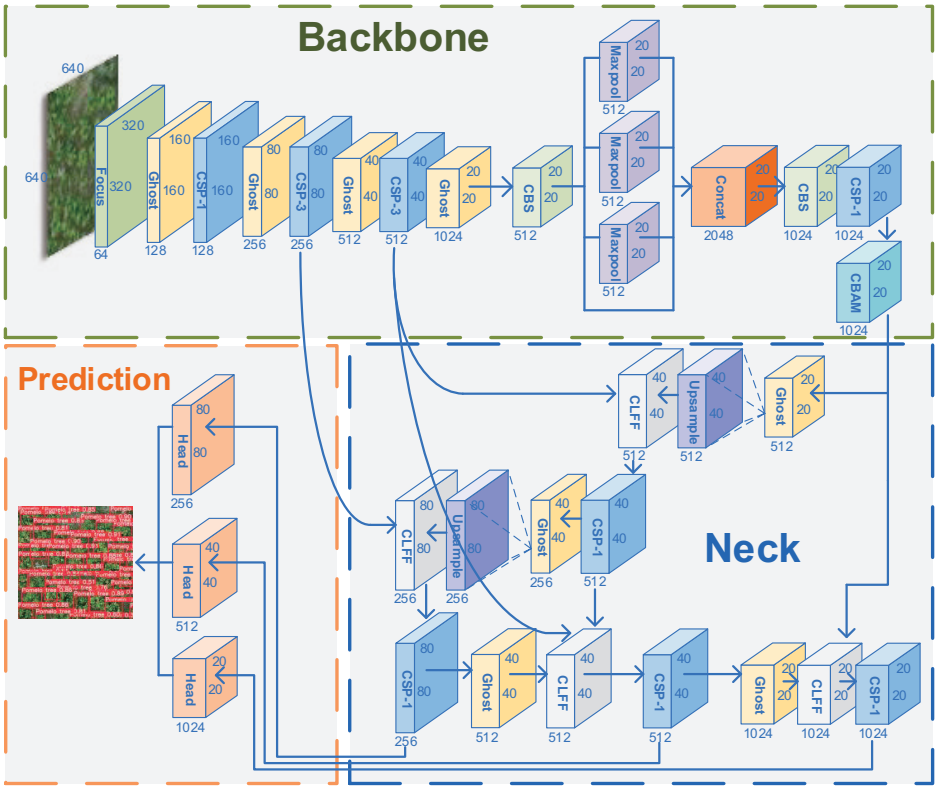


Figure 3. Proposed network structure. Backbone: Focus, CBS (convolution, batch normalization, and sigmoid weighted liner unit (SiLU) activation), Ghost, and cross-stage partial (CSP) module downsample input image and convolve data. Spatial pyramid pooling (SPP) module is embedded in last Ghost and CSP module, including three maxpooling layers and concat mode. End of backbone: convolutional block attention module (CBAM) adds weight for target information. Neck section: bidirectional feature pyramid network (BiFPN) and CLFF module transfer feature information. Three convolution sets predict class label and object location.

In the spatial attention mechanism, maximum and average pooling are applied for the feature map on the channels, and the outputs are stacked in the channel dimension to obtain an $h \times w \times 2$ feature map. The stacked feature map is fed to a convolution module, and the weight of each feature point is obtained by the sigmoid function. The input feature map is multiplied by the weight of each feature point.

The channel and spatial attention mechanisms are formulated as

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \tag{1}$$

$$M_s(F) = \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])), \tag{2}$$

where σ denotes the sigmoid function, F represents the feature map, MLP represents the fully connected neural network, and $f^{7\times7}$ represents convolution with a 7×7 filter.

In this study, the spatial attention module uses a 7×7 convolution kernel, which empirically outperforms a 3×3 convolution kernel. Within a certain interval, the larger the convolution kernel, the better the performance of the network. We maintain the original backbone feature network structure. To retain the excellent feature extraction capability of the original model, a hybrid attention mechanism is added after the backbone network.

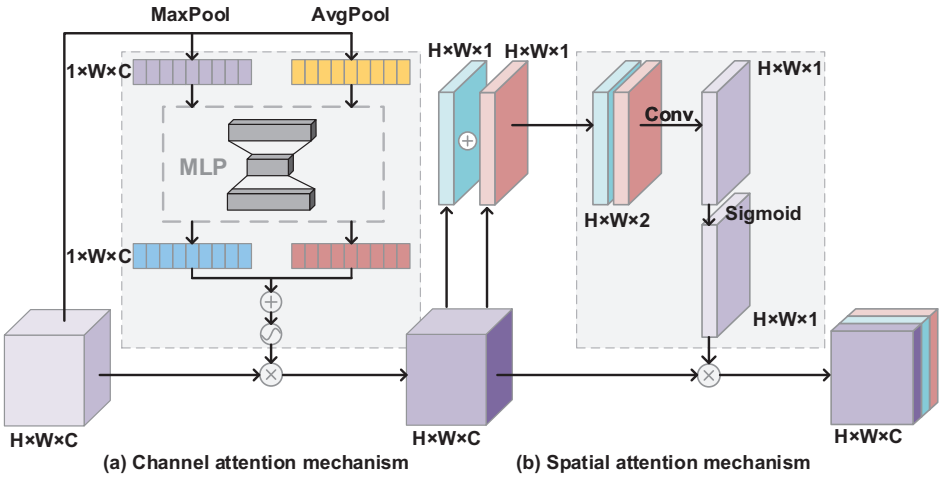


Figure 4. Hybrid attention mechanism. Maximum and average pooling are applied for the feature map to obtain channel attention as well as spatial attention.

2.2.2. Cross-Layer Feature Fusion Pyramid

The low-level features from the shallow layers of the network contain much fine-grained feature information and background noise, while features extracted from the deeper layers have more semantic information [57]. Therefore, integrating low- and high-level features can produce high-quality feature maps that complement each other. The original model feature fusion section uses a feature pyramid network (FPN) with a path-aggregation network (PANet) structure. The FPN structure conveys the deep feature information by upsampling to fuse and obtain the predicted feature map. A bottom-up feature pyramid containing two PANet structures is added after the FPN structure. The PANet can convey strongly localized features in a bottom-up manner. However, the FPN+PAN structure uses some transformations to feature maps so that their sizes are equal, which leads to the loss of some useful information. Furthermore, the FPN+PAN structure does not fully use the complementary features across the shallow and deep layers of the network, so it does not achieve better performance.

EfficientDet [58] uses BiFPN to combine different levels of features to detect objects, using features with stronger semantic information to detect large objects, and features with stronger spatial information to detect small objects. It shows good performance.

Inspired by EfficientDet, we propose the CLFF module (Figure 5) to extend the feature fusion network structure of BiFPN, which exploits the complementary features of the shallow and deep layers of the network. Unlike the aggregation strategy of series or additive operations, we consider the complementary features between different layers of the network to overcome the lack of some detailed and semantic information of deep and shallow features, respectively, about the pomelo tree. We multiply the masks of the feature maps of the middle layer with those of the shallow and deep layers to take full advantage of the complementary features between layers, which can enable one to more effectively focus on the pomelo tree region, while reducing the interference of background noise. We use the feature maps in the bottom-up path to generate the masks. The steps are as follows.

1. For the middle feature map **s**, we generate the semantic mask **w** using a convolution with a 3×3 kernel;
2. We multiply the semantic mask **w** and shallow-feature map **h**, and the semantic mask **w** and deep-feature map **g**;
3. We sum the above two results, and feed the sum to a 3×3 convolution layer to obtain the output of the feature-fusion module.

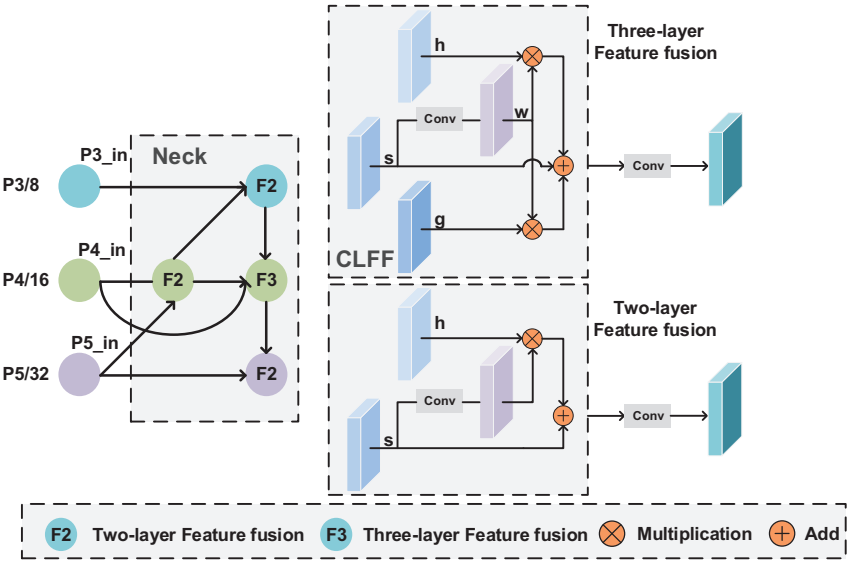


Figure 5. Neck structure and CLFF module. On the neck section, we delete low-utilization nodes and connect nodes with different layers. On CLFF, we generate the mask using the feature map of the bottom-up feature pyramid, and multiply the mask with the shallow and deep-feature maps. The two or three feature maps are fused by addition.

2.2.3. Ghost Convolution Module

The feature maps obtained by traditional convolution have high similarity and redundancy, resulting in a large computational cost. We replace this with the Ghost module to reduce the computational cost and make the model more lightweight and efficient. The Ghost module uses linear operations instead of some convolutions, as shown in Figure 6.

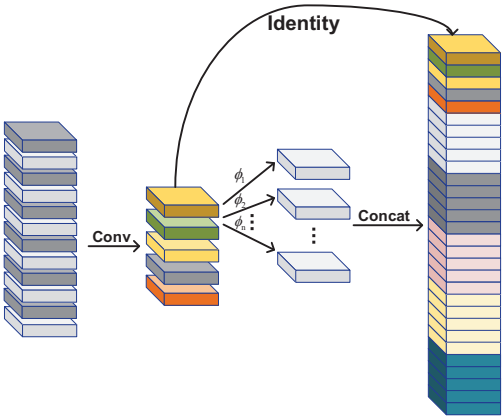


Figure 6. Ghost module. Convolution module generates intrinsic feature maps with small channels. Linear operation expands features and increases number of channels.

The Ghost module has two steps. Traditional convolution generates a small number of intrinsic feature maps, and a linear operation expands features and increases the number of channels. Linear operations can produce similar feature maps with fewer parameters and less computing cost. The total number of required parameters and the computational complexity of the Ghost module are less than those of traditional convolution.

2.3. Pomelo Tree-Detection Network

Figure 3 shows the proposed network structure for pomelo tree detection. Following the divide-and-conquer principle, module y1 focuses on large-scale object detection, and modules y2 and y3 focus on medium- and small-scale objects, respectively. A hybrid attention mechanism, a cross-layer feature-fusion pyramid, and the Ghost module improve performance.

The proposed network has a backbone, neck, and prediction sections, including a Focus module, CBS structure, CSP residual structure, and SPP module.

- The Focus module uses a slicing operation to split a high-resolution feature map into multiple low-resolution feature maps. This module samples and splices the input feature maps in each column and obtains output feature maps by convolution operations, which can reduce information loss due to downsampling;
- The CBS structure consists of convolutional layers, normalization processing, and SiLU activation functions, which have the characteristics of no upper or lower bound, smoothness, and non-monotonicity, which can improve accuracy;
- The CSP structure consists of a standard convolutional structure and a bottleneck module, which reduces and then expands the number of channels, with the final number of input and output channels remaining the same. The input feature layer of the CSP has two branches, one with multi-bottleneck stacking and standard convolution, and the other with a basic convolution module, as shown in Figure 7. The feature maps of the two branches are aggregated by a concat operation. To reduce the model size, we only stack the bottleneck modules once in the CSP structure;
- The SPP module can realize the fusion of local and global features, which enriches the information of the feature map. It performs well in the case of large differences in target size.

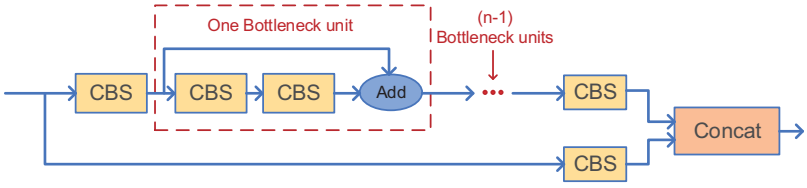


Figure 7. CSP−n structure. The input feature layer of the CSP has two branches, one with multi-bottleneck stacking and standard convolution, and the other with a basic convolution module. CSP−1 means one bottleneck unit, CSP−n means n bottleneck units.

We describe the flow of our proposed algorithm. The input of the backbone feature network is a 640 × 640 × 3 image, which is turned into a 320 × 320 × 12 feature map by the Focus module after one convolution operation with 64 convolution kernels. Through one layer of CBS and CSP modules, the shallow features are aggregated, and the feature dimension is transformed to 160 × 160 × 128, where the CBS module changes the size and number of channels of the feature map, and the CSP module divides the feature map into two parts and merges them through the cross-stage hierarchy. The features are further extracted by three CBS and CSP combination modules to obtain two effective feature layers, y1 and y2, whose respective feature maps are 80 × 80 × 256 and 40 × 40 × 512, respectively. An SPP module is inserted between the subsequent CBS and CSP structures, and the 20 × 20 × 512 feature map is fused with local and global features to improve its expressiveness. The third effective feature layer, y3, is obtained, which has a 20 × 20 × 1024 feature map.

The deep feature information enhances the network's ability to capture the target by blending the attention mechanism and fusing its weights. The cross-layer feature fusion network achieves the fusion and multiplexing of multi-level features, thus obtaining effective feature layers of the size 80 × 80 × 256, 40 × 40 × 512, and 20 × 20 × 1024. After

prediction, three results are obtained for each feature layer by decoupled head: the category (cls), coordinates (Reg), and foreground background judgment of the target frame (Obj), as shown in Figure 8. Reg has four channels, representing the offset of the center of the prediction frame compared to the feature points, and the offset of the width and height of the prediction frame compared to the logarithmic index of the reference. Obj has one channel, representing the probability of each feature point predicting the objects contained in the frame. Cls has num_classes channels, representing the probability that each feature point corresponds to a class of objects.

We use complete-IoU ($CIoU$) loss instead of intersection over union (IoU) in the prediction phase. IoU is commonly used as a matching degree evaluation metric of prediction bounding boxes and ground-truth boxes in a dataset, calculated by the ratio of their area intersection and union. We consider the effects of the overlap region, centroid distance, and aspect ratio on the loss function, which makes the regression of target-detection frames more stable. The $CIoU$ loss is defined as

$$CIoU = IoU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v \tag{3}$$

$$\alpha = \frac{v}{1 - IoU + v} \tag{4}$$

$$v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{5}$$

$$LOSS_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v, \tag{6}$$

where c is the diagonal distance of the smallest closed area that can contain both the predicted and real bounding boxes; $\rho^2(b, b^{gt})$ is the Euclidean distance between the center point of the predicted and real boxes, and the corresponding loss is $1 - CIoU$.

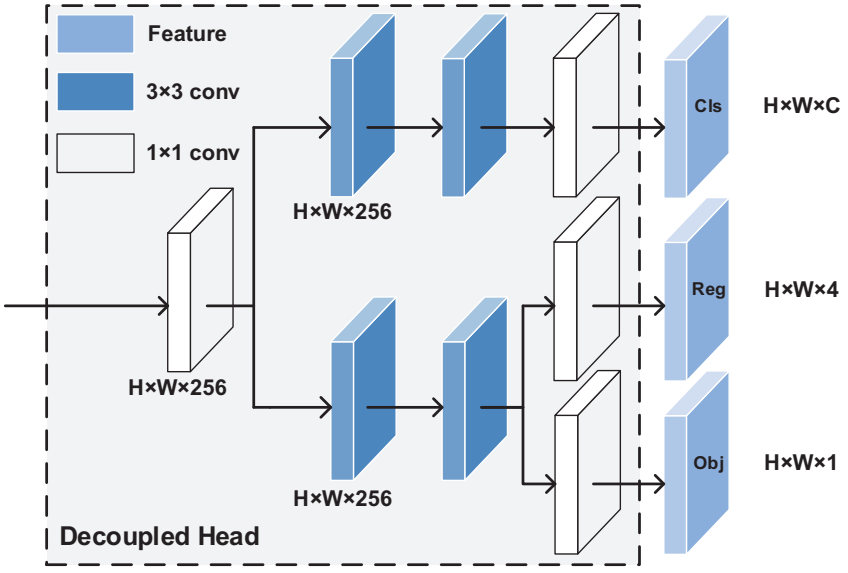


Figure 8. Decoupled head structure. For each level of neck output feature, we first adopt a 1×1 convolution layer to reduce the feature channel to 256 and then add two parallel branches with two 3×3 convolution layers each for classification and regression tasks, respectively. Obj branch is added on the regression branch.

3. Results

Table 1 shows the experimental environment. The proposed method used stochastic gradient descent (SGD) for training with 500 iterations, where the batch size was 16, the momentum coefficient was 0.937, the weight decay rate was 0.0005, and the initial learning rate was 0.01 and dynamically decreased to 0.0001. The enhancement factors of hue (H), saturation (S), and luminance (V) were set to 0.015, 0.7, and 0.4, respectively. The final output was the identified pomelo canopy location boxes and the probability of belonging to the pomelo tree category. The training, validation, and testing sets are described in Section 2.1. The source code for the proposed method is available at https://github.com/hr8yhtzb/PTDM.

Table 1. Lab environment.

Configuration	Parameter
CPU	Intel Core i9-10900kes
GPU	2 NVIDIA GeForce RTX 3090
Accelerated environment	CUDA 11.3 CUDNN8.2.1
Development	PyCharm2021.1.1
Operating system	Ubantu 18.04
Model frame	PyTorch 1.10

3.1. Standard of Performance Evaluation

We used the common index AP to evaluate the performance of different methods. Because the detection target of this study only belonged to one class, the value of mAP was equal to the single-target AP value. Hence, mAP was not used as an evaluation metric. AP was calculated as

$$AP = \int_0^1 P(R)dR, \tag{7}$$

where P and R are the respective precision and recall of the detection model,

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN}, \tag{9}$$

where TP, FP, TN, and FN denote true positive, false positive, true negative, and false negative, respectively.

The counting performance was evaluated using mean error (MAE), counting accuracy (ACC), R^2, and root mean square error ($RMSE$). MAE reflects the accuracy of counting, and $RMSE$ reflects the robustness of the counting network. They were defined as

$$MAE = \frac{1}{n}\sum_1^n |t_i - c_i| \tag{10}$$

$$ACC = (1 - \frac{1}{n}\sum_1^n \frac{|t_i - c_i|}{t_i}) \tag{11}$$

$$R^2 = 1 - \frac{\sum_1^n (t_i - c_i)^2}{\sum_1^n (t_i - \widetilde{t}_i)^2} \tag{12}$$

$$RMSE = \sqrt{\frac{\sum_1^n (t_i - c_i)^2}{n}}, \tag{13}$$

where t_i, $\widetilde{t}_i$, and c_i are the actual count, average true count, and predicted count, respectively, of image i (total number of anchors detected), and n is the number of UAV images in the test set. MAE and ACC quantify prediction accuracy. R^2 and $RMSE$ were used to evaluate the counting performance of the proposed method.

3.2. Comparison to State-of-the-Art Object-Detection Algorithms

We compare our proposed method with state-of-the-art object-detection algorithms, including Faster R-CNN [37], SSD [43], YOLOv3 [45], YOLOv4-tiny [46], Libra [42], and CCTD [39]. Faster R-CNN is a famous two-stage detection algorithm, and many tree-detection models are based on Faster R-CNN. SSD, YOLOv3, and YOLOv4-tiny are famous single-stage detection algorithms. They do not have the bounding box proposal and resampling steps, so they have a faster computational speed. YOLOv4-tiny is a lightweight version of YOLOv4 with fewer parameters and faster detection speed. Libra and CCTD are state-of-the-art two-stage detection algorithms. Libra optimized two-stage detection using IoU-balanced sampling. CCTD used a multi-level region proposal network to optimize the selection of region proposals. The two datasets described in Section 2.1 were selected for experiments.

3.2.1. Comparison of Detection Performance

We first evaluated the precision, recall, AP, F1-score and complexity for different state-of-the-art algorithms, as shown in Tables 2–4, where the best value of different methods is shown in bold, from which we can observe the following.

- Faster R-CNN had the lowest precision, with just 43.17% and 16.08%, respectively, in datasets 1 and 2, perhaps because of the complex background. Faster R-CNN does not build an image feature pyramid, and cannot effectively use shallow and small-scale features, resulting in a high number of false detections and low precision. In addition, this method appears to overfit, which resulted in much lower accuracy than other methods, indicating that this method is unsuitable for pomelo tree detection.
- SSD had an extremely low recall, with 58.23% and 30.06% in datasets 1 and 2, respectively, because SSD has no feature pyramid, the same as in Faster R-CNN. The recall rate of SSD was 1% to 4% lower than that of Faster R-CNN, which uses two-step detection. It first generates the region of interest, and then detects within it. Therefore, two-step detection could reduce the number of missed objects, and had a higher recall rate. However, the recall rates of SSD and Faster R-CNN were both lower than those of other methods owing to the lack of a feature pyramid.
- YOLOv3 had the highest precision of all methods, reaching over 93% in the first dataset and 91% in the second region. However, its recall was less than 80% and 50% in the two datasets, respectively. YOLOv3 is the most complex because it includes a large number of convolution modules, which incur more computational cost.
- YOLOv4-tiny had similar detection results to YOLOv3, as they are both single-stage detectors. Although YOLOx-nano is also a single-stage detector, it had about 6% to 25% higher recall than YOLOv4-tiny and YOLOv3 in both regions because it has two PANet structures that can constitute a bottom-up feature pyramid, which can enhance feature extraction. In addition, YOLOx-nano is anchor-free, which is better than an anchor-based detector for single-tree detection in remote-sensing images [59]. Because an anchor-based detector matches the object based on the anchor box's size, it misses detection if the object's size exceeds that of the anchor box. The anchor free detector efficiently eliminates the problem that the anchor box does not match the object size and lowers the possibility of missed detection.
- Libra and CCTD are both two-stage detectors and therefore have a high recall rate on both datasets, with about 87% in the first dataset and 80% in the second dataset. This result indicated that Libra and CCTD method had fewer missed detections. However, because Libra and CCTD are anchor-based method, their accuracy is limited, with only about 63% to 75% in the second dataset.
- Our method obtained the highest AP among all algorithms, which demonstrates its effectiveness. The precision was 92.41% and 87.18% in datasets 1 and 2, respectively, the recall was 87.07% and 75.35%, and the AP value was 93.74% and 87.81%. Among all compared methods, the AP value of ours was the highest. The outstanding performance of our method can be attributed to the attention mechanism, the cross-layer

feature-fusion pyramid, and the Ghost module. The attention mechanism improves the capacity to extract feature information across space and channels, and provides enough feature suppression background information. The cross-layer feature-fusion pyramid combines semantic information from feature maps at different levels of layers, allowing it to learn rich information. Use of the Ghost module instead of 3×3 convolution reduces the variance of the feature geometry, thus deepening the feature information association between deep and shallow feature maps.

- The model size of our proposed method was 7.8 MB only, which is 98% and 96% smaller than that of Libra and YOLOv3, respectively, and is just slightly more than that of YOLOx-nano. In addition, our method was the fastest of all methods. It is worth noting that the size of our proposed method is larger than YOLOx-nano, but it runs faster than YOLOx-nano. This is because the ghost module we used can reduce the computational complexity. In summary, our improvements make the model lighter and more computationally efficient.

Table 2. Comparison of state-of-the-art object-detection algorithms in dataset 1.

Algorithm	Precision (%)	Recall (%)	AP (%)	F1-Score
Faster R-CNN	43.17	63.99	53.98	0.52
SSD	82.56	58.23	68.92	0.68
YOLOv3	**93.64**	79.24	91.74	0.86
YOLOv4-tiny	89.93	81.26	89.53	0.85
YOLOx-nano	90.99	86.43	93.08	0.89
Libra	87.12	**87.85**	89.25	0.87
CCTD	87.29	87.64	91.61	0.87
ours	92.41	87.07	**93.74**	**0.90**

Table 3. Comparison of state-of-the-art object-detection algorithms in dataset 2.

Algorithm	Precision (%)	Recall (%)	AP (%)	F1-Score
Faster R-CNN	16.08	30.16	8.99	0.21
SSD	87.91	30.06	60.26	0.45
YOLOv3	**91.81**	46.72	73.31	0.62
YOLOv4-tiny	84.73	62.37	79.09	0.72
YOLOx-nano	84.41	71.09	83.66	0.77
Libra	63.98	72.22	69.25	0.68
CCTD	76.67	**83.84**	84.72	0.80
ours	87.18	75.35	**87.81**	**0.81**

Table 4. Comparison of computational complexity.

Algorithms	Model Size (MB)	The Average Detection Time	The Shortest Detection Time
Faster R-CNN	107.86	0.262 s	0.248 s
SSD	90.07	0.159 s	0.125 s
YOLOv3	234.69	0.196 s	0.174 s
YOLOv4-tiny	22.41	0.133 s	0.119 s
YOLOx-nano	**2.7**	0.133 s	0.121 s
Libra	466	0.872 s	0.828 s
CCTD	315	0.615 s	0.588 s
ours	7.8	**0.099 s**	**0.091 s**

Figure 9 shows the visual detection effects of different methods. Faster R-CNN and SSD had a large number of missed detections, whereas YOLOv3, YOLOv4-tiny, Libra, and CCTD had a small number. This result showed that the recalls of the Faster R-CNN and SSD detectors were much lower than those of the YOLO series because they lack an image pyramid and are unable to properly integrate the information from the feature layer. Libra

and CCTD are both anchor-based detection methods, and they are insensitive to tiny targets. Overall, our method outperformed the other methods, without the yellow box and the blue box for the testing image. Moreover, as a lightweight model, our method is better suited for pomelo tree recognition.

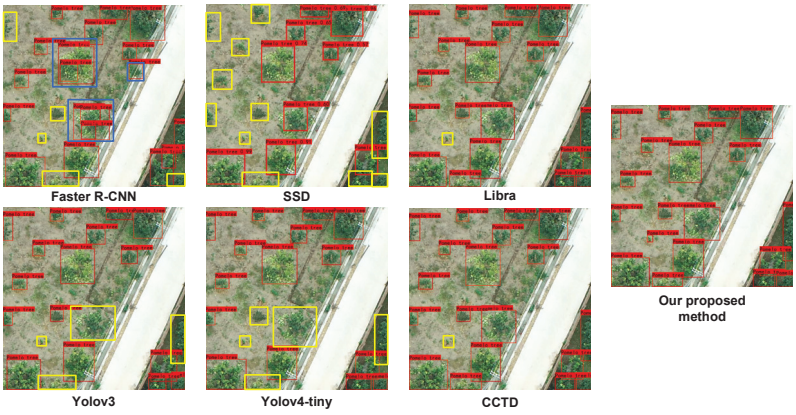


Figure 9. Detection effects of different methods (yellow box: missed detection; blue box: error detection). It can be seen that Faster R-CNN and SSD without feature pyramids had a large number of missed detections. YOLO series with feature pyramids had a small number of missed detections. Libra and CCTD could not detect tiny targets. The proposed method had no missed detection.

Figure 10 shows the ability of the proposed method to distinguish similar targets. In Figure 10a,b, an area planted with a large number of pomelo trees is confused with a small number of orange trees. The proposed method could accurately treat orange trees with slightly different leaf colors as negative samples. In Figure 10c, the proposed method could accurately treat other trees with mostly the same leaf color but with a few white leaves as negative samples. Overall, our proposed method had good ability to distinguish trees similar to pomelo trees.

Figure 10. The ability of the proposed method to distinguish different citrus trees (yellow box: other citrus fruit tree). In (**a**,**b**) the proposed method distinguishes orange trees well. In (**c**), the method can accurately distinguish other trees with slightly different leaf colors. It can be seen that proposed method had a good ability to distinguish similar targets.

3.2.2. Counting Performance

We evaluated the detection of the number of pomelo trees for different state-of-the-art methods, choosing 15 images of pomelo orchards captured by a UAV as the dataset. Each image contained 504 to 1490 pomelo trees, in terrain types of flat plains and uneven mountains, with both dense and sparse distributions of pomelo trees. The comparison results are shown in Table 5. The *MAE* and *RMSE* of the proposed method were 36.4

and 42.7, respectively, significantly better compared with Faster R-CNN, SSD, YOLOv3, YOLOv4-tiny, CTDD, and Libra. In particular, the *MAE* and *RMSE* of the proposed method were significantly better than those of YOLOv3, with improvements of 39.6% and 42.2%, respectively. Therefore, the proposed method can better extract features of different scales, and can deal with multi-scale changing scenes as well as negative samples.

Table 5. Tree counting performance for different methods.

Algorithm	MAE	RMSE	ACC (%)	R^2
Faster R-CNN	170.9	221.2	82.18	0.24
SSD	162.7	184.7	82.61	0.47
YOLOv3	60.3	73.9	92.91	0.92
YOLOv4-tiny	59.9	73.5	93.12	0.92
Libra	55.5	67.1	92.97	0.93
CCTD	56.1	77.9	93.07	0.91
ours	**36.4**	**42.7**	**95.93**	**0.97**

Figure 11 compares the *MAE* and *RMSE* counting results for different methods. The *MAE* and *RMSE* of Faster R-CNN were the highest, with 170.9 and 221.2, respectively. This result confirmed that Faster R-CNN performs poorly in two locations for detection. SSD's *MAE* and *RMSE* were much higher than those of YOLOv3 and YOLOv4-tiny and only slightly lower than those of Faster R-CNN. This is because SSD lacks the feature pyramid structure, which prevents the method from extracting sufficient features to identify pomelo tree. The *MAE* and *RMSE* of YOLOv3, YOLOv4-tiny, Libra, and CCTD were similar, with *MAE* fluctuating between 55 and 60 and *RMSE* between 65 and 80. The *MAE* and *RMSE* of our proposed method were lower than those of other methods, with 36.4 and 42.7, which indicates its advantages in terms of computational accuracy and robustness.

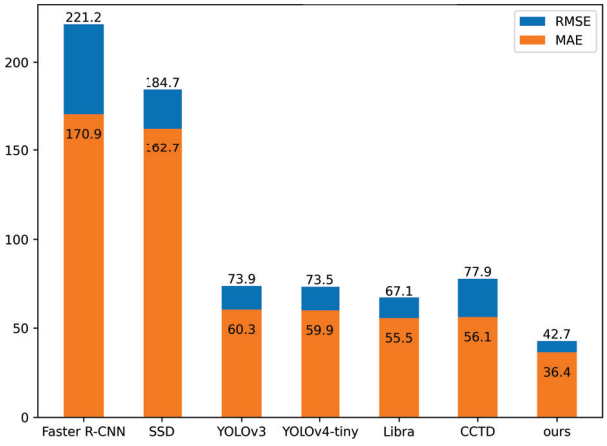


Figure 11. Comparison of *MAE* and *RMSE* counting results for different methods. The proposed method has the lowest *MAE* and *RMSE*.

Figure 12 illustrates the predicted counts of the proposed method and true counts of the 15 images, including the predicted counts obtained by the result of detection, and the ground truth counts. For almost all images, the proposed method predicted counts that were extremely near to ground-truth box counts for all images. The errors between the predicted counts and the true counts were from 7 to 71. In the UAV images with large errors, the orchards are complex and contain many additional plants, such as bushes. The number of false detections increased because they were mistakenly identified as pomelo trees. Even yet, the accuracy of prediction exceeded 95%, which indicates the proposed

method produces a reasonable estimate of the number of pomelo trees. Note that most images in the test data set contain over 900 pomelo trees, and the counting results here are the correctly detected fruit trees.

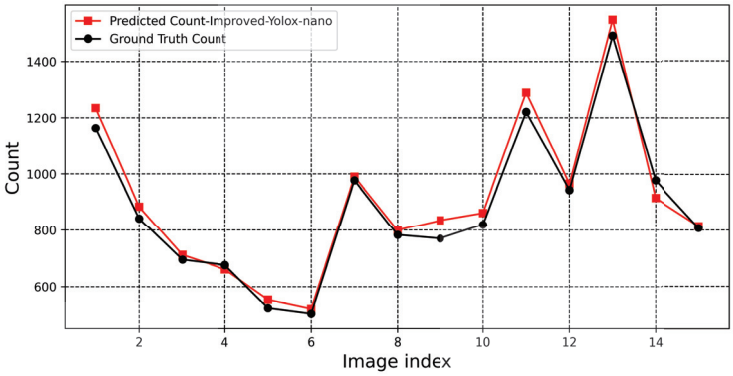


Figure 12. Predicted counts of proposed method and true counts of 15 images.

3.3. Ablation Experiments

Through ablation experiments, we could analyze the impact of different components on the proposed method. We chose *AP*.5 and *AP*.5:.95 as assessment indicators after testing the model's performance with several modules. *AP*.5 was the AP value when *IoU* was taken as 0.5. *AP*.5:.95 is the *AP* value when *IoU* increased from 50% to 95% in steps of 5%. *AP*.5 could reflect the performance of model detection, and *AP*.5:.95 could reflect the robustness of model-detection performance. The experimental results in Table 6 show that our method (last row) significantly improved the detection effect compared with the original method.

Table 6. Comparison of components of proposed method.

YOLOx-Nano	CBAM	CLFF	Ghost	*AP*.5 (%)	*AP*.5:.95 (%)
√				93.08	61.0
√	√			93.38	61.1
√		√		93.21	60.9
√			√	93.36	61.1
√	√	√		93.53	61.3
√	√	√	√	**93.74**	**61.5**

3.3.1. Attention Mechanism

When we added a hybrid attention mechanism module at the end of the backbone feature extraction network, *AP* increased from 93.08% to 93.38%. Owing to the relatively simple backbone structure of the original lightweight network, it performed poorly when the background and target were not sufficiently distinguished. The hybrid attention mechanism module weights the pixels of the feature map with channel attention and spatial attention, which can improve the ability of feature extraction and effectively highlight pomelo tree regions over backgrounds.

3.3.2. Use of Cross-Layer Fusion Feature Pyramid

When we added the cross-layer fusion feature module, *AP* improved from 93.08% to 93.21%. This indicates that the CLFF module improves detection performance because it utilizes complementary characteristics between the extracted shallow detail information and deep semantic information.

3.3.3. Use of Ghost Module

When we used the Ghost module instead of traditional convolution for feature extraction, AP increased from 93.08% to 93.36% because the Ghost module can obtain a large number of redundant feature maps with a simple linear operation. In addition, the Ghost module could reduce the effect of feature geometry variation, reduce the parameters and computational complexity of the deep network, and extract more effective feature information. It is worth noting that when the Ghost, CBAM, and CLFF modules were combined, the AP improved from 93.08% to 93.74%, which indicates that the proposed method is more capable of feature extraction and has better detection performance than the original method.

3.3.4. Visual Effect

Figure 13 compares the original YOLOx-nano and the proposed method. From Figure 13a,d, we can find that the pomelo trees partially obscured by wires are not recognized by the original method, while the proposed method successfully identifies them, which indicates our method's better robustness against and recognition of obscured objects.

According to Figure 13b,e, the young pomelo trees had small canopies that the original method could not recognize. In addition, the original method ignored two closely adjacent pomelo trees, treating them as negative samples with no obvious boundary. In contrast, the proposed method could accurately identify small objects and pomelo trees with inconspicuous edges. The original method expands the perceptual field under the layer-by-layer convolution, which ignores some small pomelo trees. The proposed method adds an attention mechanism and a cross-layer feature-fusion mechanism, making it more capable of identifying small targets and targets with unclear edges.

According to Figure 13c,d, the original method incorrectly identified a banana tree as a pomelo tree, while the proposed method avoided this error. This is because the original method has fewer parameters in the backbone feature extraction network, and the extracted features are insufficient. The proposed method adds an attention mechanism, improving feature fusion and making the scale of feature differentiation between positive and negative samples more obvious.

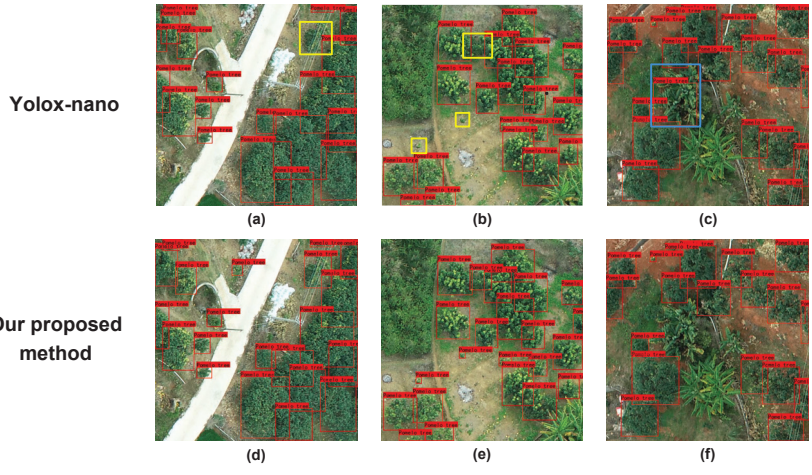


Figure 13. Comparison of the detection effect between the original and proposed methods (yellow box: missed detection; blue box: error detection): (**a**) the original method missed a pomelo tree obscured by power lines; (**b**) the original method missed pomelo trees with small canopies and inconspicuous canopy boundaries; (**c**) the original method incorrectly treated banana trees as pomelo trees; (**d**–**f**) the proposed method avoids all the above errors to accurately identify all pomelo trees.

4. Discussion

We proposed a pomelo tree-detection method for UAV remote-sensing images. We introduced a hybrid attention mechanism module to improve the ability of feature extraction and effectively highlight pomelo tree regions over backgrounds. We designed a feature-fusion module to fuse feature maps of the same scale but different levels, without greatly increasing computation. We replaced the convolution module with a Ghost module to improve model detection. The proposed method reduces model parameters while extracting more effective feature information. Compared with some state-of-the-art target-detection algorithms, our method experimentally showed better detection performance and fewer parameters, so it is better suited for pomelo tree detection in UAV images.

In our future work, we will research how to use domain adaption to detect pomelo trees according to a different time and space, and extend our proposed method to different types of trees in orchards.

Author Contributions: Conceptualization, H.Y. and K.H.; Data curation, H.Y., K.H. and Y.X.; Formal analysis, H.Y., K.H., Y.X. and J.D.; Funding acquisition, K.H., J.D. and Z.Y.; Investigation, H.Y., K.H., C.R., Y.X. and J.D.; Methodology, H.Y., K.H. and C.R.; Project administration, K.H. and Z.Y.; Resources, H.Y., K.H. and Y.X.; Software, H.Y. and K.H.; Supervision, K.H., C.R., J.D. and Z.Y.; Validation, H.Y., K.H., C.R., J.D. and Z.Y.; Visualization, H.Y., K.H. and J.D.; Writing—original draft, H.Y. and K.H.; Writing—review & editing, H.Y., K.H., C.R., Y.X., J.D. and Z.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China under Grants 61976104, 61906046, and 61976229, the Natural Science Foundation of Guangdong Province under Grant 2020A1515010702, the Guangdong Province Special Project in Key Fields for Universities under Grant 2020ZDZX3044, the Open Research Projects of Zhejiang Lab under Grant 2021KH0AB08 and the Science and Technology Program of Guangdong Province under Grant 2020B121201013: Guangdong Provincial Key Laboratory of Conservation and Precision Utilization of Characteristic Agricultural Resources in Mountainous Areas.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

ACC	Accuracy
BiFPN	Bidirectional Feature Pyramid Network
CBMA	Convolutional Block Attention Module
CBS	Convolution, Batch normalization and SiLU activation
$CIoU$	Complete-IoU
CLFF	Cross-Layer Feature Fusion
CNNs	Convolutional Neural Networks
CSP	Cross-Stage Partial
IOU	Intersection Over Union
FPN	Feature Pyramid Network
MAE	Mean Error
PANet	Path Aggregation Network
RMSE	Root Mean Square Error
RPN	Region Proposal Network
R^2	Correlation Coefficient
SGD	Stochastic Gradient Descent
SiLU	Sigmoid Weighted Liner Unit
SPP	Spatial Pyramid Pooling
UAVs	Unmanned Aerial Vehicles

References

1. Morton, J.F. *Fruits of Warm Climates*; JF Morton: Miami, FL, USA, 1987.
2. Jiménez-Brenes, F.M.; López-Granados, F.; De Castro, A.; Torres-Sánchez, J.; Serrano, N.; Peña, J. Quantifying pruning impacts on olive tree architecture and annual canopy growth by using UAV-based 3D modelling. *Plant Methods* **2017**, *13*, 55. [CrossRef] [PubMed]
3. Castillo-Ruiz, F.J.; Jimenez-Jimenez, F.; Blanco-Roldán, G.L.; Sola-Guirado, R.R.; Agueera-Vega, J.; Castro-Garcia, S. Analysis of fruit and oil quantity and quality distribution in high-density olive trees in order to improve the mechanical harvesting process. *Span. J. Agric. Res.* **2015**, *13*, e0209. [CrossRef]
4. Garcia-Ruiz , F.; Sankaran, S.; Maja, J.M.; Lee, W.S.; Rasmussen, J.; Ehsani, R. Comparison of two aerial imaging platforms for identification of Huanglongbing-infected citrus trees. *Comput. Electron. Agric.* **2013**, *91*, 106–115. [CrossRef]
5. Zhang, C.; Valente, J.; Kooistra, L.; Guo, L.; Wang, W. Orchard management with small unmanned aerial vehicles: A survey of sensing and analysis approaches. *Precis. Agric.* **2021**, *22*, 2007–2052. [CrossRef]
6. Barbagallo, S.; Consoli, S.; Russo, A. A one-layer satellite surface energy balance for estimating evapotranspiration rates and crop water stress indexes. *Sensors* **2009**, *9*, 1–21. [CrossRef] [PubMed]
7. Salgadoe, A.S.A.; Robson, A.J.; Lamb, D.W.; Dann, E.K.; Searle, C. Quantifying the severity of phytophthora root rot disease in avocado trees using image analysis. *Remote Sens.* **2018**, *10*, 226. [CrossRef]
8. Moran, M.S.; Inoue, Y.; Barnes, E. Opportunities and limitations for image-based remote sensing in precision crop management. *Remote Sens. Environ.* **1997**, *61*, 319–346. [CrossRef]
9. Wal, T.; Abma, B.; Viguria, A.; Prévinaire, E.; Zarco-Tejada, P.J.; Serruys, P.; Valkengoed, E.V.; Voet, P. Fieldcopter: Unmanned aerial systems for crop monitoring services. In *Precision Agriculture '13*; Wageningen Academic Publishers: Wageningen, The Netherlands, 2013; pp. 169–175.
10. Ochoa, K.S.; Guo, Z. A framework for the management of agricultural resources with automated aerial imagery detection. *Comput. Electron. Agric.* **2019**, *162*, 53–69. [CrossRef]
11. Swetnam, T.L.; Falk, D.A. Application of metabolic scaling theory to reduce error in local maxima tree segmentation from aerial LiDAR. *For. Ecol. Manag.* **2014**, *323*, 158–167. [CrossRef]
12. Yang, J.; He, Y.; Caspersen, J.P.; Jones, T.A. Delineating individual tree crowns in an uneven-aged, mixed broadleaf forest using multispectral watershed segmentation and multiscale fitting. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *10*, 1390–1401. [CrossRef]
13. Jing, L.; Hu, B.; Noland, T.; Li, J. An individual tree crown delineation method based on multi-scale segmentation of imagery. *ISPRS J. Photogramm. Remote Sens.* **2012**, *70*, 88–98. [CrossRef]
14. Srestasathiern, P.; Rakwatin, P. Oil palm tree detection with high resolution multi-spectral satellite imagery. *Remote Sens.* **2014**, *6*, 9749–9774. [CrossRef]
15. Dos Santos, A.M.; Mitja, D.; Delaître, E.; Demagistri, L.; de Souza Miranda, I.; Libourel, T.; Petit, M. Estimating babassu palm density using automatic palm tree detection with very high spatial resolution satellite images. *J. Environ. Manag.* **2017**, *193*, 40–51. [CrossRef] [PubMed]
16. Pu, R.; Landry, S. A comparative analysis of high spatial resolution IKONOS and WorldView-2 imagery for mapping urban tree species. *Remote Sens. Environ.* **2012**, *124*, 516–533. [CrossRef]
17. Hung, C.; Bryson, M.; Sukkarieh, S. Multi-class predictive template for tree crown detection. *ISPRS J. Photogramm. Remote Sens.* **2012**, *68*, 170–183. [CrossRef]
18. Dalponte, M.; Ørka, H.O.; Ene, L.T.; Gobakken, T.; Næsset, E. Tree crown delineation and tree species classification in boreal forests using hyperspectral and ALS data. *Remote Sens. Environ.* **2014**, *140*, 306–317. [CrossRef]
19. López-López, M.; Calderón, R.; González-Dugo, V.; Zarco-Tejada, P.J.; Fereres, E. Early detection and quantification of almond red leaf blotch using high-resolution hyperspectral and thermal imagery. *Remote Sens.* **2016**, *8*, 276. [CrossRef]
20. Nevalainen, O.; Honkavaara, E.; Tuominen, S.; Viljanen, N.; Hakala, T.; Yu, X.; Hyyppä, J.; Saari, H.; Pölönen, I.; Imai, N.N.; et al. Individual tree detection and classification with UAV-based photogrammetric point clouds and hyperspectral imaging. *Remote Sens.* **2017**, *9*, 185. [CrossRef]
21. Wang, Y.; Zhu, X.; Wu, B. Automatic detection of individual oil palm trees from UAV images using HOG features and an SVM classifier. *Int. J. Remote Sens.* **2019**, *40*, 7356–7370. [CrossRef]
22. Huang, K.K.; Ren, C.X.; Liu, H.; Lai, Z.R.; Yu, Y.F.; Dai, D.Q. Hyperspectral image classification via discriminant Gabor ensemble filter. *IEEE Trans. Cybern.* **2021**, *52*, 8352–8365. [CrossRef] [PubMed]
23. Albetis, J.; Duthoit, S.; Guttler, F.; Jacquin, A.; Goulard, M.; Poilvé, H.; Féret, J.B.; Dedieu, G. Detection of Flavescence dorée grapevine disease using unmanned aerial vehicle (UAV) multispectral imagery. *Remote Sens.* **2017**, *9*, 308. [CrossRef]
24. Lei, S.; Luo, J.; Tao, X.; Qiu, Z. Remote Sensing Detecting of Yellow Leaf Disease of Arecanut Based on UAV Multisource Sensors. *Remote Sens.* **2021**, *13*, 4562. [CrossRef]
25. Zhang, Y.; Wa, S.; Liu, Y.; Zhou, X.; Sun, P.; Ma, Q. High-Accuracy Detection of Maize Leaf Diseases CNN Based on Multi-Pathway Activation Function Module. *Remote Sens.* **2021**, *13*, 4218. [CrossRef]
26. Nofrizal, A.Y.; Sonobe, R.; Yamashita, H.; Seki, H.; Mihara, H.; Morita, A.; Ikka, T. Evaluation of a One-Dimensional Convolution Neural Network for Chlorophyll Content Estimation Using a Compact Spectrometer. *Remote Sens.* **2022**, *14*, 1997. [CrossRef]

27. Milioto, A.; Lottes, P.; Stachniss, C. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2229–2235.
28. Potena, C.; Nardi, D.; Pretto, A. Fast and accurate crop and weed identification with summarized train sets for precision agriculture. In Proceedings of the International Conference on Intelligent Autonomous Systems, Shanghai, China, 3–7 July 2016; pp. 105–121.
29. Milella, A.; Marani, R.; Petitti, A.; Reina, G. In-field high throughput grapevine phenotyping with a consumer-grade depth camera. *Comput. Electron. Agric.* **2019**, *156*, 293–306. [CrossRef]
30. Qi, X.; Dong, J.; Lan, Y.; Zhu, H. Method for Identifying Litchi Picking Position Based on YOLOv5 and PSPNet. *Remote Sens.* **2022**, *14*, 2004. [CrossRef]
31. Huang, B.; Zhao, B.; Song, Y. Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote-sensing imagery. *Remote Sens. Environ.* **2018**, *214*, 73–86. [CrossRef]
32. Huang, K.K.; Ren, C.X.; Liu, H.; Lai, Z.R.; Yu, Y.F.; Dai, D.Q. Hyperspectral image classification via discriminative convolutional neural network with an improved triplet loss. *Pattern Recognit.* **2021**, *112*, 107744. [CrossRef]
33. Li, W.; Fu, H.; Yu, L.; Cracknell, A. Deep learning based oil palm tree detection and counting for high-resolution remote-sensing images. *Remote Sens.* **2016**, *9*, 22. [CrossRef]
34. Pibre, L.; Chaumon, M.; Subsol, G.; Lenco, D.; Derras, M. How to deal with multi-source data for tree detection based on deep learning. In Proceedings of the 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, Canada, 14–16 November 2017; pp. 1150–1154.
35. Wu, J.; Yang, G.; Yang, H.; Zhu, Y.; Li, Z.; Lei, L.; Zhao, C. Extracting apple tree crown information from remote imagery using deep learning. *Comput. Electron. Agric.* **2020**, *174*, 105504. [CrossRef]
36. Zheng, J.; Fu, H.; Li, W.; Wu, W.; Yu, L.; Yuan, S.; Tao, W.Y.W.; Pang, T.K.; Kanniah, K.D. Growing status observation for oil palm trees using Unmanned Aerial Vehicle (UAV) images. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 95–121. [CrossRef]
37. 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]
38. Osco, L.P.; De Arruda, M.d.S.; Junior, J.M.; Da Silva, N.B.; Ramos, A.P.M.; Moryia, É.A.S.; Imai, N.N.; Pereira, D.R.; Creste, J.E.; Matsubara, E.T.; et al. A convolutional neural network approach for counting and geolocating citrus-trees in UAV multispectral imagery. *ISPRS J. Photogramm. Remote Sens.* **2020**, *160*, 97–106. [CrossRef]
39. Zheng, J.; Wu, W.; Yu, L.; Fu, H. Coconut Trees Detection on the Tenarunga Using High-Resolution Satellite Images and Deep Learning. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 6512–6515.
40. Zheng, J.; Wu, W.; Yuan, S.; Fu, H.; Li, W.; Yu, L. Multisource-domain generalization-based oil palm tree detection using very-high-resolution (vhr) satellite images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
41. Zheng, J.; Fu, H.; Li, W.; Wu, W.; Zhao, Y.; Dong, R.; Yu, L. Cross-regional oil palm tree counting and detection via a multi-level attention domain adaptation network. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 154–177. [CrossRef]
42. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra r-cnn: Towards balanced learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 821–830.
43. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
44. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
45. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
46. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-yolov4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13029–13038.
47. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
48. Liu, R.; Tao, F.; Liu, X.; Na, J.; Leng, H.; Wu, J.; Zhou, T. RAANet: A Residual ASPP with Attention Framework for Semantic Segmentation of High-Resolution Remote Sensing Images. *Remote Sens.* **2022**, *14*, 3109. [CrossRef]
49. Han, Z.; Hu, W.; Peng, S.; Lin, H.; Zhang, J.; Zhou, J.; Wang, P.; Dian, Y. Detection of Standing Dead Trees after Pine Wilt Disease Outbreak with Airborne Remote Sensing Imagery by Multi-Scale Spatial Attention Deep Learning and Gaussian Kernel Approach. *Remote Sens.* **2022**, *14*, 3075. [CrossRef]
50. Gong, H.; Mu, T.; Li, Q.; Dai, H.; Li, C.; He, Z.; Wang, W.; Han, F.; Tuniyazi, A.; Li, H.; et al. Swin-Transformer-Enabled YOLOv5 with Attention Mechanism for Small Object Detection on Satellite Images. *Remote Sens.* **2022**, *14*, 2861. [CrossRef]
51. Li, X.; Pan, J.; Xie, F.; Zeng, J.; Li, Q.; Huang, X.; Liu, D.; Wang, X. Fast and accurate green pepper detection in complex backgrounds via an improved Yolov4-tiny model. *Comput. Electron. Agric.* **2021**, *191*, 106503. [CrossRef]
52. Yu, J.; Wu, T.; Zhou, S.; Pan, H.; Zhang, X.; Zhang, W. An SAR Ship Object Detection Algorithm Based on Feature Information Efficient Representation Network. *Remote Sens.* **2022**, *14*, 3489. [CrossRef]
53. Cheng, G.; Yang, C.; Yao, X.; Guo, L.; Han, J. When deep learning meets metric learning: Remote-sensing image scene classification via learning discriminative CNNs. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 2811–2821. [CrossRef]

54. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1580–1589.
55. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
56. Li, M.; Zhai, Y.M.; Luo, Y.W.; Ge, P.F.; Ren, C.X. Enhanced transport distance for unsupervised domain adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 13936–13944.
57. Liu, J.J.; Hou, Q.; Cheng, M.M.; Feng, J.; Jiang, J. A simple pooling-based design for real-time salient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3917–3926.
58. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10781–10790.
59. Zamboni, P.; Junior, J.M.; Silva, J.d.A.; Miyoshi, G.T.; Matsubara, E.T.; Nogueira, K.; Gonçalves, 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]

remote sensing

MDPI

Article

Can Machine Learning Algorithms Successfully Predict Grassland Aboveground Biomass?

Yue Wang [1], Rongzhu Qin [1], Huzi Cheng [2], Tiangang Liang [3], Kaiping Zhang [1], Ning Chai [1], Jinlong Gao [3], Qisheng Feng [3], Mengjing Hou [3], Jie Liu [3], Chenli Liu [3], Wenjuan Zhang [4], Yanjie Fang [5], Jie Huang [6] and Feng Zhang [1,7,*]

1 College of Ecology, Lanzhou University, Lanzhou 730000, China
2 Laboratory for the Cognitive Control, Department of Psychological and Brain Sciences, Indiana University Bloomington, Bloomington, IN 47401, USA
3 State Key Laboratory of Grassland Agro-Ecosystems, College of Pastoral Agriculture Science and Technology, Lanzhou University, Lanzhou 730020, China
4 Institute of Qinghai Provincial Natural Resources Survey and Monitoring, Xining 810000, China
5 Key Laboratory of High Water Utilization on Dryland of Gansu Province, Institute of Dryland Farming, Gansu Academy of Agricultural Sciences, Lanzhou 730070, China
6 Animal Husbandry, Pasture and Green Agriculture Institute, Gansu Academy of Agricultural Sciences, Lanzhou 730000, China
7 NAU-MSU Asia Hub, Nanjing Agricultural University, Nanjing 210095, China
* Correspondence: zhangfeng@lzu.edu.cn

Abstract: The timely and accurate estimation of grassland aboveground biomass (AGB) is important. Machine learning (ML) has been widely used in the past few decades to deal with complex relationships. In this study, based on an 11-year period (2005–2015) of AGB data (1620 valid AGB measurements) on the Three-River Headwaters Region (TRHR), combined with remote sensing data, weather data, terrain data, and soil data, we compared the predictive performance of a linear statistical method, machine learning (ML) methods, and evaluated their temporal and spatial scalability. The results show that machine learning can predict grassland biomass well, and the existence of an independent validation set can help us better understand the prediction performance of the model. Our findings show the following: (1) The random forest (RF) based on variables obtained through stepwise regression analysis (SRA) was the best model (R^2_{vad} = 0.60, $RMSE_{vad}$ = 1245.85 kg DW (dry matter weight)/ha, AIC = 5583.51, and BIC = 5631.10). It also had the best predictive capability of years with unknown areas (R^2_{indep} = 0.50, $RMSE_{indep}$ = 1332.59 kg DW/ha). (2) Variable screening improved the accuracy of all of the models. (3) All models' predictive accuracy varied between 0.45 and 0.60, and the RMSE values were lower than 1457.26 kg DW/ha, indicating that the results were reliably accurate.

Keywords: MODIS; Google Earth Engine; biomass inversion; spatio-temporal scalability; model building

Citation: Wang, Y.; Qin, R.; Cheng, H.; Liang, T.; Zhang, K.; Chai, N.; Gao, J.; Feng, Q.; Hou, M.; Liu, J.; et al. Can Machine Learning Algorithms Successfully Predict Grassland Aboveground Biomass? *Remote Sens.* **2022**, *14*, 3843. https://doi.org/10.3390/rs14163843

Academic Editors: Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Received: 18 June 2022
Accepted: 4 August 2022
Published: 9 August 2022

1. Introduction

Grassland is one of the most widespread types of vegetation in the world, and it accounts for about 20% of the global land area. It plays an important role in ecological balance and human livelihood [1]. The aboveground biomass (AGB) of grassland is one of the most direct manifestations of grassland quality and grassland ecosystems [2,3]. Therefore, accurate estimation of the grassland AGB is particularly important for grassland grazing management and regional grassland sustainable development.

The aboveground biomass (AGB) can be predicted by direct methods (by harvesting the biomass) and by indirect methods (including the use of remote sensing tools). The direct harvest method has high estimation accuracy, but it is time-consuming, labor-intensive, costly, and inefficient, and will cause a certain degree of damage to grassland ecology [4].

Therefore, it is only suitable for short-term, small-scale detection. In contrast, Satellite remote sensing has low cost and high efficiency, providing an effective means for regional and global production detection [5]. Enhanced vegetation index (EVI) [6], soil adjusted vegetation index (SAVI) [7,8], modified soil adjustment vegetation index (MSAVI) [9], and ratio vegetation index (RVI) [10] have been used for monitoring and estimation. Furthermore, other environmental factors that affect biomass (such as climate variables and soil properties) contain non-biological information [11–13].

The use of remote sensing images and environmental factors to construct non-parametric models is a common method for estimating grassland biomass. The construction of a non-parametric model requires a "learning process" based on training data that can automatically optimize the weight of each calculation until error has been minimized [14]. Non-parametric models can be divided into linear and non-linear models. Classical linear models include partial least squares (PLS) and principal component regression (PCR). Common non-linear models include machine learning (ML) models, such as convolutional neural networks (CNNs), support vector machines (SVMs), and random forests (RFs).

Grassland growth can be influenced by multiple environmental factors, and previous studies have suggested that estimating AGB use with only a single type of factor could introduce errors and uncertainties [15]. Although ML-based simulations of grasslands using different algorithms yield different accuracies [3], in general, machine learning still outperforms traditional algorithms in terms of simulating grasslands due to its strong interpretability and high efficiency [16]. ML methods, such as random forest (RF) regression, can integrate multiple factors and learn highly complex nonlinear mappings for estimating AGB. Xie et al. used Landsat data to establish artificial neural network (ANN) and multiple linear regression (MLR) models to estimate the grassland AGB in Inner Mongolia (n = 461) [16]. The results show that compared to MLR (RMSEr = 49.51% for the training, and RMSEr = 53.20% for the testing), ANN (RMSEr = 39.88% for the training, and RMSEr = 42.36% for the testing) can provide more accurate results. Tang et al. established a RF algorithm suitable for the Headwater of the Yellow River ($R^2_{val} = 0.56$, $RMSE_{val} = 51.3\ g/m^2$) [15]. Many studies have been conducted on grasslands, however, the small number of available samples and the lack of support from long-term observational data persist as challenges [17].

In recent decades, many vegetation indices have been used to estimate AGB, such as the NDVI [18–21]. However, the variation in the AGB is not influenced by a single factor, but by a variety of factors, such as the soil, climate, and topography. Some simple vegetation indices can help in understanding the effect of explanatory variables on biomass availability but may not be able to describe the biological processes that occur in nature. Therefore, this study hopes to combine soil, climate, topography, remote sensing, and other factors with machine learning to better predict grassland biomass.

The main objectives of this research are to (1) compare the ability of linear regression models and machine learning algorithms to evaluate grassland biomass using years of continuous observations and (2) evaluate spatio-temporal scalability between the traditional methods and machine learning-based methods. This paper is organized as follows. Section 2 describes the data sources and methods. Section 3 compares the model accuracy and spatio-temporal scalability, and inverts the aboveground biomass of grassland in the Three-River Headwaters Region (TRHR) based on the optimal results. In Section 4, the distribution pattern of grassland biomass, the spatio-temporal scalability of the model, the input variables that affect grassland biomass, and the factors that affect the accuracy of the model are discussed. Conclusions are summarized in Section 5.

2. Data Sources and Methods

2.1. Data Sources

2.1.1. Study Area

The study area (31°39′~36°12′N, 89°45′~102°23′E) is located in the southern part of Qinghai Province in China. It is the ecological barrier between the roofs of the world (the

Qinghai Tibet Plateau) and is the headwater source of the three largest rivers in China: the Yellow River, the Yangtze River, and the Lancang River. The TRHR is known as the China Water Tower and provides a barrier for environmental protection and sustainable development for the middle and lower reaches of rivers in China and Southeast Asian countries. The study area has a total area of 36,561,502 ha, accounting for about 43% of the total area of Qinghai Province. The average altitude is 4000 m, the annual mean temperature (AMT) is 3 °C, the annual mean precipitation (AMP) is 377 mm, and its range of growing degree days (GDDs) is 0–5001 °C. The grassland in the TRHR is dominated by alpine meadows and alpine grasslands, accounting for 54% and 16% of the area, respectively (Figure 1b). The soil distribution in the area has prominent vertical zoning rules, mainly alpine meadow soil and swamp meadow soil, and the frozen soil layer is well developed [22]. Details of research sites are supplemented in the Table S1.

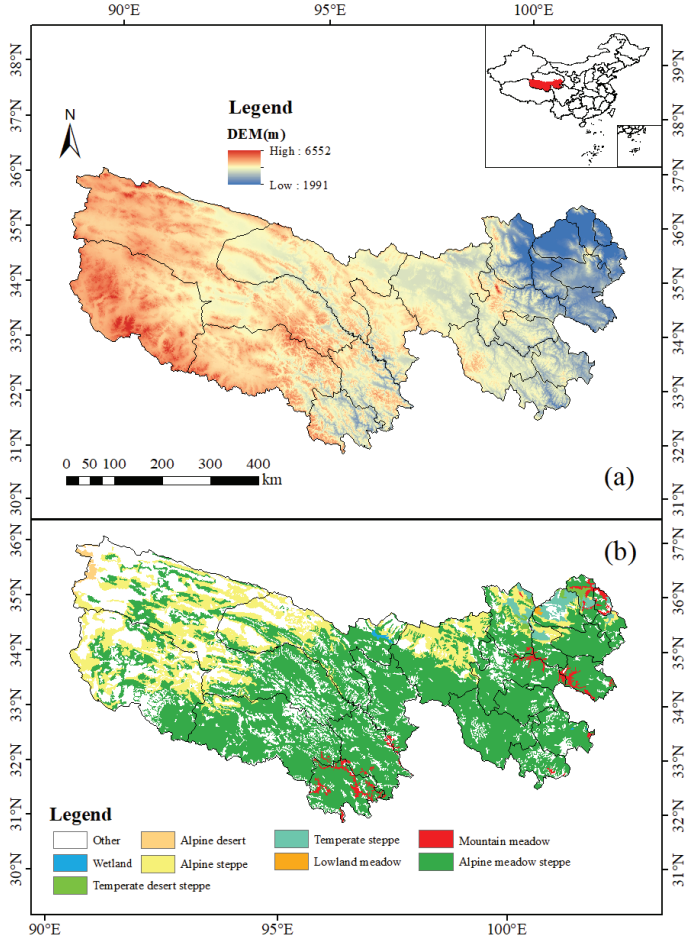


Figure 1. Digital elevation model (DEM) (**a**) and spatial distribution patterns of grassland type (**b**) in the pastoral area of Southern Qinghai Province, China.

2.1.2. AGB Dataset

We collected field survey AGB data during the peak growing season (July to September) from 2005 to 2015, for a total of 1620 valid data items (Table S2). Guide, Guinan, Jianzha, and Tongren were newly added in 2015, and each county has only 2 to 5 sample points.

The largest number of samples was of those drawn from Banma county, with 120 sample points collected, followed by Tongde county, with 115 sample points collected.

The general spatial distribution of AGB measurements during 2005–2015 provides an overall picture of the AGB values of the study area (Figure 2). As shown in the figure, the value of AGB was in the range from 200 to 10,000 kg DW (dry matter weight)/ha (the average value was 3090 kg DW/ha). The figure shows an overall downward trend from southeast to northwest, with some exceptions.

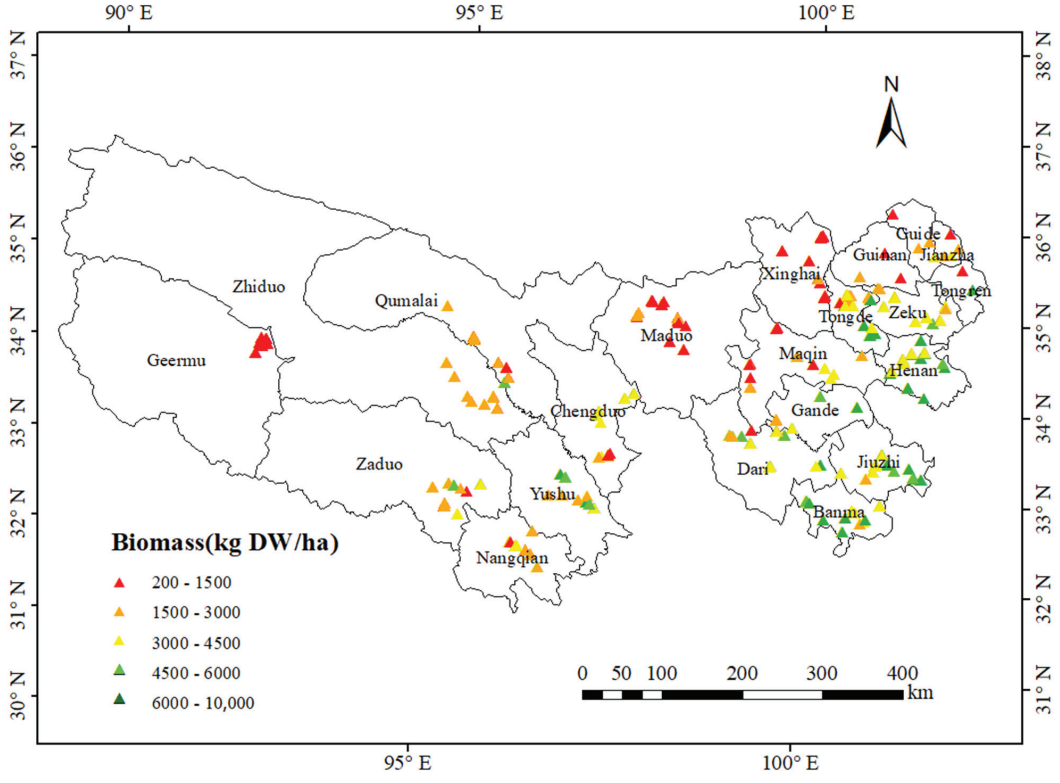


Figure 2. Grassland AGB measured from 2005 to 2015 (n = 1620) (all AGB values measured at each observation sample station are averaged for that observation station).

We now outline the methodological steps undertaken to collect the grassland AGB of the study (Figure S1).

a) The latitude and longitude of the TRHR were determined by a handheld GPS device.
b) We established a grassland sample plot (500 m × 500 m) based on typical grassland vegetation communities that had a relatively flat terrain and uniform growth and that were spatially representative. We used five 1 m × 1 m grassland observation plots in the sample plot using the five point method.
c) The aboveground part of the vegetation in each observation plot was mowed up to the ground. All litter and other non-plant materials were removed from the grass samples, bagged, and brought back to the laboratory for further processing.
d) We weighed samples from each plot in the laboratory. They were then oven-dried at 65 °C for 48 h, and their dry weights were recorded.

All AGB values (dry weight) in a MODIS pixel (500 × 500 m) were averaged to represent the average AGB of the MODIS pixels, and the center latitude and longitude of the pixel were used for modeling.

2.1.3. Meteorological Data

Climatic data as an environmental factor are the basis of research fields such as agriculture and forestry. We collected the daily maximum temperature, minimum temperature, and precipitation data from 15 meteorological stations in the TRHR from 2005 to 2015 from the China Meteorological Data Network (http://data.cma.cn, accessed on 3 February 2021). AMT and AMP were interpolated by ANUSPLIN, an interpolation package specially designed for meteorological data [23].

2.1.4. Soil Data and Topographic Data

The soil data were from the global gridded soil information (https://soilgrids.org/, accessed on 3 February 2021) and included the organic carbon stock of soil (OC) on the surface (0–5 cm), organic carbon density (OR) on the surface (0–5 cm), bulk density (BL) of the soil surface (0–5 cm), (CL) of the soil surface (0–5 cm), coarse fragments (CR) (0–5 cm), silt size (SL) of the soil surface (0–5 cm), sand (SN) on the soil surface (0–5 cm), cation exchange capacity (at pH = 7) (CE) of the soil surface (0–5 cm), and pH water (pH) in the soil surface (0–5 cm). We then resampled the data to 500 m.

The digital elevation model (DEM) data were obtained from Shuttle Radar Topography Mission (SRTM) images (version 004) (http://srtm.csi.cgiar.org, accessed on 3 February 2021). To match the available data, the digital elevation data were resampled to 500 m, and the projection type was defined as a WGS_1984 map projection. In addition, ArcGIS software was used to generate the aspect and slope with a resolution of 90 m; the data were then resampled to 500 m. Finally, we extracted the corresponding data and analyzed them.

2.1.5. MODIS Data and Its Processing

All MODIS data in this paper were obtained from the Google Earth Engine (GEE) platform (https://code.earthengine.google.com/, accessed on 7 February 2021) (version 006) (Tables 1 and S3). The processing flowchart is shown in Figure S2.

Table 1. MODIS products.

MODIS	Time Resolution (d)	Spatial Resolution (m)	Bands
MOD09A1	8	500	B1–B7
MOD13A1	16	500	NDVI, EVI
MOD11A2	8	1000	D-LST, N-LST
MOD15A2H	8	500	LAI, Fpar

Note: B1–B7: reflectance of MODIS bands 1–7; NDVI: normalized difference vegetative index; EVI: enhanced vegetation index; the unit of the day land surface temperature (D-LST) and night land surface temperature (N-LST) is K; LAI: leaf area index; Fpar: fraction of photosynthetically active radiation.

2.2. Method and Modeling

2.2.1. Variable Selection

Three variable selection methods, stepwise regression analysis (SRA), ridge regression (RR), and the least absolute shrinkage and selection operator (LASSO), were used in this study. As a filter of variable indicators, SRA can quickly select the most important variable indicators related to the research object from a large number of indicator libraries [24]. RR is a variable screening method and has the ability to handle multicollinearity data [25]. LASSO can automatically select the most important independent variables and narrow down the less important predictor variables to zero [26].

2.2.2. Summary of Modeling Methods

The PLS, SVM, RF, Gradient Boosting Decision Trees (GBDT), and Multilayer BP Neural Network (BP) modeling methods were used. The PLS is a mathematical regression model that determines the correlation between variables [27]. The two most important parameters in the RF algorithm are the number of regression trees and the number of predictors at each node. When the number of regression trees is set larger, the accuracy

of the model will also be improved, but the model operation time will be prolonged. The default value of the number of predictors at each node is 1/3 of the total number of independent variables [28]. The SVM is a type of machine learning theory based on statistical learning theory [29]. In this paper, the radial basis function was used as the kernel function, and the genetic algorithm was used to optimize two key parameters (gamma and cost). These three algorithms use functions from the R packages "PLSR", "random forest", and "e1071." GBDT is an integrated model based on a decision tree that contains flexible and efficient machine learning algorithms [30]. We continuously optimized the three hyperparameters of the learning rate, the number of iterations, and the subsample. The GBDT method was implemented based on the gradient boosting regressor in the sklearn package. BP is a multi-layer forward neural network, and its theoretical basis is the error direction propagation algorithm [31]. The most important parameters in the BP model are the number of neurons and hidden layers, which need to be repeatedly tested and continuously tuned. The BP model is built based on the torch deep learning framework. The rationale for machine learning algorithms was added to the Supplementary Materials (Text S1).

2.2.3. Assessing Model Accuracy

The square of the correlation coefficient between the measured value on the ground and the predicted value of the model (R^2) and root mean square error (RMSE) values were used as the standards of accuracy evaluation. Higher R^2 and lower RMSE indicate better model performance. Equations (1) and (2) express R^2 and RMSE respectively:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Yi - \hat{Y}i)^2}{\sum_{i=1}^{n}(Yi - \overline{Yi})^2} \quad (1)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Yi - \hat{Y}i)^2}{n}} \quad (2)$$

where Yi represents the measured value of the aboveground biomass of grassland, $\hat{Y}i$ is the predicted value of Yi, and $\overline{Yi}$ is its average value.

The model selection process took into account its fitting performance and simplicity. In this study, AIC and BIC were used as the evaluation criteria. Among the models with the same fitting ability, the model with the smaller BIC was preferred.

Equations (3) and (4) express AIC and BIC respectively:

$$AIC = k\ \ln(n) + n\ \ln\left(\frac{\sum_{i=1}^{n}(Yi - \hat{Y}i)^2}{n}\right) \quad (3)$$

$$BIC = 2k + n\ \ln\left(\frac{\sum_{i=1}^{n}(Yi - \hat{Y}i)^2}{n}\right) \quad (4)$$

where Yi represents the measured value of the aboveground biomass of grassland, $\hat{Y}i$ is the predicted value of Yi, k represents the number of variables in the model, and n represents the number of samples. The larger the R^2, the higher the credibility of the model prediction, the smaller the RMSE, AIC, and BIC, the better the model fitting effect.

3. Results

3.1. Correlation between Grassland AGB and Variables

This study used the correlation analysis method to test the correlation between AGB and MODIS data, topographical factors, soil factors, and meteorological factors (Table S4). As shown in the table, there was a significant correlation between the AGB and most MODIS vegetation indices. Among them, grassland AGB had the highest positive correlation coefficient with the NDVI, MSAVI, optimized soil-adjusted vegetation index (OSAVI),

and SAVI (R = 0.51). The correlation coefficient between AGB and the reflectance of the MODIS bands (B1–B7) was between −0.44 and 0.39. The correlation with B7 was the largest (R = −0.44). There was a strong correlation between the aboveground biomass of grassland and the five band indices (C–G). Among them, AGB had the largest correlation with E.

The AGB was significantly correlated with most variables, and only had a weak relationship with topographical factors and SL among soil factors (R < 0.1). AGB and slope had the highest correlation coefficient (R = 0.29), followed by the DEM, and the weakest relationship was with the aspect. The BLD, SN, and pH were negatively correlated with the AGB, but the AGB had a positive correlation with the other soil factors, among which, the relationship with CL was the strongest (R = 0.26). Among the meteorological factors, AGB had a significant relationship with AMT, GDD, and AMP, but only showed a negative correlation with GDD.

3.2. Variable Screening and Model Evaluation

We divided the variables into the All set (45 variables), SRA subset (12 variables), RR subset (11 variables), and LASSO subset (17 variables) (Table 2). We used these variable sets as input variables and respectively constructed the PLS, RF, SVM, GBDT, BP models, for a total of 20 models. The accuracy of the predicted aboveground biomass of each model in the TRHR was assessed (Table 3). The results show the following:

(1) Overall, the R^2 of the training set (R^2_{train}) of the 20 models was between 0.35 and 0.94, with an average of 0.67, and the RMSE of the training set ($RMSE_{train}$) was between 460.09 and 1499.63 kg DW/ha, with an average of 1045.87 kg DW/ha. The R^2 of the validation set (R^2_{vad}) was between 0.45 and 0.6, with an average of 0.53, and the RMSE of the validation set ($RMSE_{vad}$) was between 1239.59 and 1457.26 kg DW/ha, with an average of 1341.46 kg DW/ha. The R^2 of the independent verification set (R^2_{indep}) was between 0.26 and 0.50, with an average of 0.38, and the RMSE of the independent verification set ($RMSE_{indep}$) was between 1332.59 and 1663.55 kg DW/ha, and the average was 1475.05 kg DW/ha. The AIC of the independent verification set was between 5583.51 and 5757.92, and the BIC of the independent verification set was between 5631.1 and 5936.4. The SRA-RF model had the largest R^2_{vad} and R^2_{indep}, the smallest $RMSE_{vad}$, $RMSE_{indep}$, AIC, and BIC, and the best predictions (RF-R^2_{vad} = 0.60, RF-$RMSE_{vad}$ = 1245.85 kgDW/ha, RF-R^2_{indep} = 0.50, RF-$RMSE_{indep}$ = 1332.59 kg DW/ha, RF-AIC = 5583.51, RF-BIC = 5631.1). The RF model based on SRA achieved more accurate prediction results with a small number of variables, so the RF-SRA (RF-R^2_{vad} = 0.60 (Figure 3a); RF-R^2_{indep} = 0.50 (Figure 3b)) was the best model.

(2) During the selection of variables, the DEM among terrain-related factors, the pH among soil-related factors, the B6 among remote sensing-related factors, and the GDD among meteorological factors were selected. These four variables had significant effects on the grassland biomass.

(3) Although the overall fitting performance of the estimation model based on the RF method (the average of RF-R^2_{train} was 0.91) was much higher than that based on the PLS method (the average of PLS-R^2_{train} was 0.36)), its predictive performance (RF-R^2_{vad} was between 0.58 and 0.6, and the average was 0.59) was not (RF-R^2_{vad} was between 0.45 and 0.50, and the average was 0.48).

(4) Judging from the prediction results of the model, among the results based on different variables, the results of the RF algorithm were superior to the other algorithms; the model had a higher R^2 and a lower RMSE (RF-All-R^2_{vad} = 0.59, RF-SRA-R^2_{vad} = 0.60, RF-RR-R^2_{vad} = 0.58, and RF-LASSO-R^2_{vad} = 0.58).

(5) Overall, the R^2_{vad} (between 0.45 and 0.6 and the average value of 0.53) and $RMSE_{vad}$ (the average value was 1341.46 kg DW/ha) of the 20 models' test sets were superior to R^2_{indep} (between 0.26 and 0.5, the average value was 0.38) and $RMSE_{indep}$ (the average value was 1475.05 kg DW/ha). Of the 20 models, 12 AGB models had values of R^2_{vad} greater than or equal to the average R^2_{vad} (R^2 = 0.53) of all models. This shows that at least 60% of the 20 models had a high accuracy and that these models can reflect 53–60% of the changes in the grassland AGB. Of the 20 models, 11 AGB models had an R^2_{indep} greater than or equal to the average R^2_{indep} (0.38) of all models, which shows that, when these models were expanded in time and space, their predictive ability declined. Of the 20 models, at least 56% reflected 38–50% of the changes in AGB in the next two years and over more space in the TRHR.

(6) We found that the model was optimal for the following combinations: (1) RF, SVM, BP, and SRA; (2) PLS, GBDT, and RR; and the model's spatio-temporal scalability was optimal for the following combinations: (3) PLS, RF, and SRA, (4) SVM, BP, and RR, (5) GBDT and LASSO. The All set had the worst performance of the models for grassland aboveground bio-mass, and variable selection helped improve model accuracy.

Table 2. Results of variables screening by different screening methods.

Methods	Variable Set	Filter Number
ALL	DEM Slope Aspect BLD CEC CL SN SL pH OR OC CR B1-B7 C D E F G BI DVI EVI Fpar LAI MSAVI NDSI NDVGI NDVI NDWI OSAVI RVI SATVI SAVI SCI TVI D-LST N-LST AMT GDD AMP	45
SRA	DEM CL pH OR OC B1 B5 B6 OSAVI D-LST N-LST GDD	12
RR	DEM SN SL pH OC B3 B5 B6 BI D-LST GDD	11
LASSO	DEM Slope CL SN pH B2 B6 C E EVI LAI MSAVI NDVGI OSAVI AMT GDD AMP	17

Note: DEM: digital elevation model; BLD: bulk density; CEC: cation exchange capacity (at pH = 7); CL: clay content; SN: sand; SL: silt size; OR: organic carbon density; OC: soil organic carbon stock; CR: coarse fragments; B1–B7: the reflectance of the MODIS bands 1–7; C–G: five band indices (Band 2–7 (C), Band 5/Band 2 (D), (Band 5 − Band 7)/(Band 5 + Band 7) (E), Band 7/Band 2 (F), and Band 7/Band 5 (G)); BI: brightness index; DVI: difference vegetation index; EVI: enhanced vegetation index; Fpar: fraction of photosynthetically active radiation; LAI: leaf area index; MSAVI: modified soil-adjusted vegetation index; NDSI: normalized difference soil index; NDVGI: normalized difference vegetation green index; NDVI: normalized difference vegetative index; NDWI: normalized difference water index; OSAVI: optimized soil-adjusted vegetation index; RVI: ratio vegetation index; SATVI: soil-adjusted total vegetation index; SAVI: soil-adjusted vegetation index; SCI: soil color index; TVI: transformed vegetation index; D-LST: day land surface temperature; N-LST: night land surface temperature; AMT: annual mean temperature; GDD: growing degree days; AMP: annual mean precipitation.

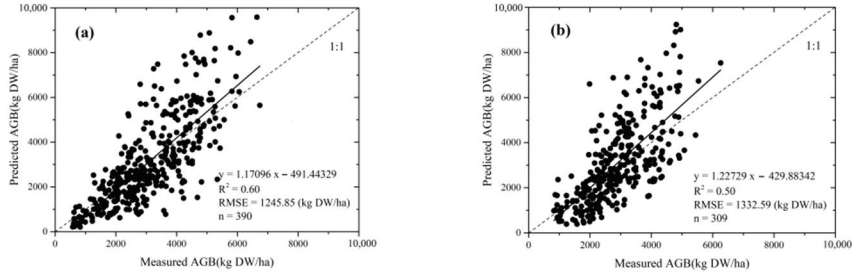


Figure 3. (**a**) Relationship between the measured biomass by RF-SRA_{vad} and that predicted by it. (**b**) Relationship between the biomass measured using the RF-SRA_{indep} and that predicted by it.

Table 3. Assessment of accuracy of the multi-factor grassland AGB estimation model.

Variable Set	Model	Training Dataset		Testing Dataset		Independent Testing Dataset		AIC	BIC
		R^2	RMSE	R^2	RMSE	R^2	RMSE		
ALL	PLS	0.38	1459.00	0.45	1431.62	0.34	1487.92	5757.92	5936.40
	RF	0.92	620.99	0.59	1253.24	0.43	1396.90	5654.12	5832.59
	SVM	0.73	1037.59	0.55	1342.84	0.41	1492.87	5707.99	5886.46
	GBDT	0.85	766.67	0.59	1239.59	0.38	1460.16	5645.58	5824.06
	BP	0.94	460.09	0.50	1427.26	0.34	1614.86	5755.54	5934.01
SRA	PLS	0.36	1484.46	0.49	1385.01	0.36	1474.83	5666.10	5713.70
	RF	0.91	664.34	0.60	1245.85	0.50	1332.59	5583.51	5631.10
	SVM	0.51	1336.17	0.54	1365.37	0.39	1490.64	5654.96	5702.56
	GBDT	0.77	931.60	0.56	1288.12	0.38	1447.90	5609.53	5657.13
	BP	0.69	1054.13	0.53	1359.04	0.30	1612.39	5651.34	5698.93
RR	PLS	0.35	1493.14	0.50	1382.86	0.35	1477.88	5662.89	5706.52
	RF	0.90	682.57	0.58	1271.56	0.46	1363.59	5597.44	5641.07
	SVM	0.48	1362.57	0.50	1399.81	0.41	1479.49	5672.39	5716.02
	GBDT	0.77	929.48	0.57	1275.69	0.41	1418.99	5599.97	5643.60
	BP	0.58	1204.95	0.49	1407.71	0.34	1515.10	5676.78	5720.41
LASSO	PLS	0.35	1499.63	0.48	1406.04	0.36	1480.31	5687.86	5755.28
	RF	0.91	657.03	0.58	1263.89	0.45	1378.47	5604.72	5672.15
	SVM	0.58	1258.28	0.55	1354.10	0.36	1503.76	5658.50	5725.92
	GBDT	0.70	1050.85	0.57	1272.33	0.41	1408.85	5609.91	5677.33
	BP	0.74	963.93	0.46	1457.26	0.26	1663.55	5715.77	5783.19

Note: SRA: stepwise regression analysis; RR: ridge regression; LASSO: least absolute shrinkage and selection operator; PLS: partial least squares; RF: random forest; SVM: support vector machine; GBDT: gradient boosting decision tree; BP: multi-layer back-propagation neural network.

The relationship between the number of sampling points and the accuracy of the model is shown in Figure 4. In general, the RF algorithm delivered the best performance, with a value of R^2 that was higher than the other algorithms. The simulation accuracy of the model changed drastically with the number of samples. We take the RF-SRA model as an example. The R^2_{vad} of the RF-SRA model was between 0.52 and 0.74, with a difference of 0.22. The slope of the trend line about the RF-SRA model was −0.0231. The R^2_{indep} of the RF-SRA model was between 0.4 and 0.5, with a difference of 0.1.

In Figure 4 the ordinate represents R^2, and the abscissa represents the sample size (30% (390 data items) meaning that 30% of all samples in 2005–2013 were randomly selected for three to seven points and were then modeled and verified); 40% means that 517 data items were used, 50% means 650 items, 60% means 794 items, 70% means 921 items, 80% means 1042 items, 90% means 1178 items, and 100% means 1311 items.

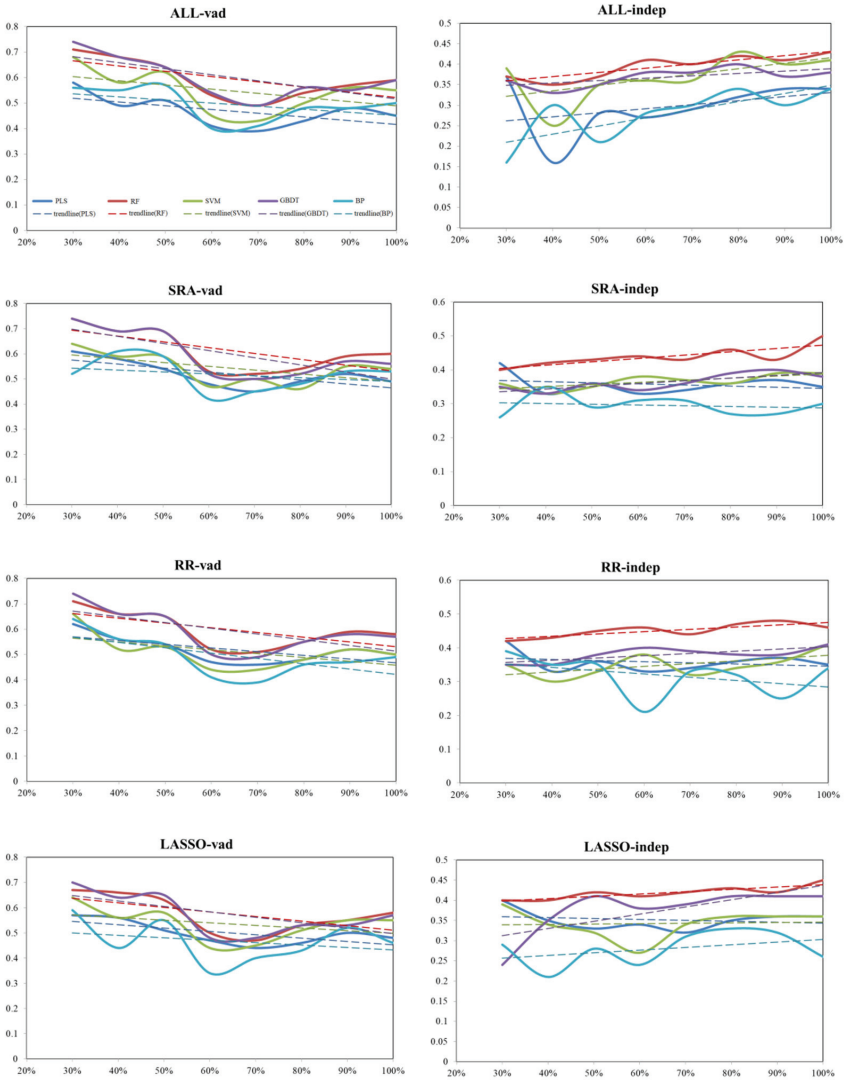
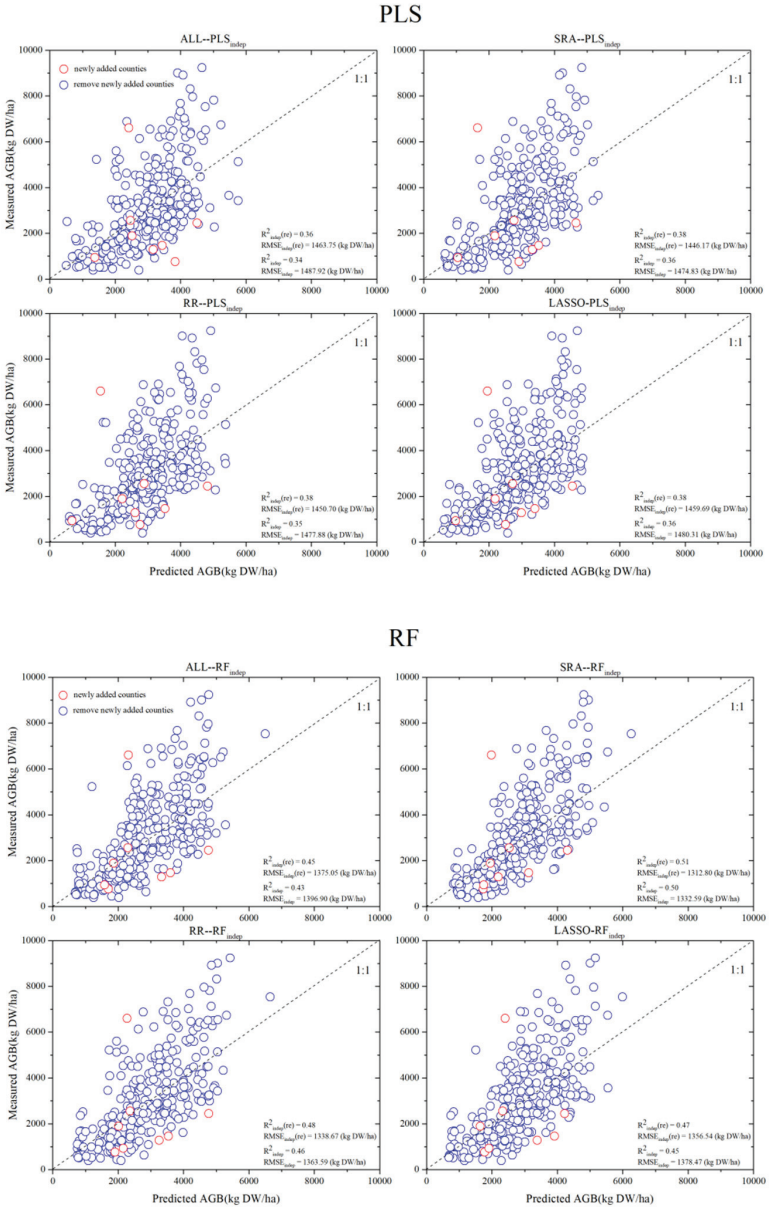


Figure 4. The relationship between changes in the number of samples of each model and R^2.

3.3. Assessing Spatial and Temporal Sample Distributions

When the five models were expanded in space and time, their accuracy decreased (the average R^2 of the 20 models decreased by 0.15 (Table 3). We bring the results (R^2_{indep}, $RMSE_{indep}$) of the independent testing dataset in Table 3 into Figure 4 for further exploration. In Figure 4, the red dots represent the AGB of the newly added locations from 2014 to 2015, the blue dots represent those of no newly added locations from 2014 to 2015, that is, "re". Comparing the R^2_{indep}, $RMSE_{indep}$ (the model with the independent validation set as the test set) and R^2_{indep} (re), $RMSE_{indep}$ (re) (the model with data that only scales in the temporal direction as the test set), the results show that the accuracy of adding new points (R^2_{indep}) was lower than that of not adding new points (R^2_{indep} (re)), which indicates that adding new points reduced accuracy. That is, when the model was extended in space, its accuracy decreased (models calibrated at small scales, when transferred to large scales, incur errors).


PLS
ALL--PLS indep
SRA--PLS indep
RR--PLS indep
LASSO-PLS indep
newly added counties
remove newly added counties
1:1
Measured AGB(kg DW/ha)
Predicted AGB(kg DW/ha)
RF
ALL--RF indep
SRA--RF indep
RR--RF indep
LASSO-RF indep

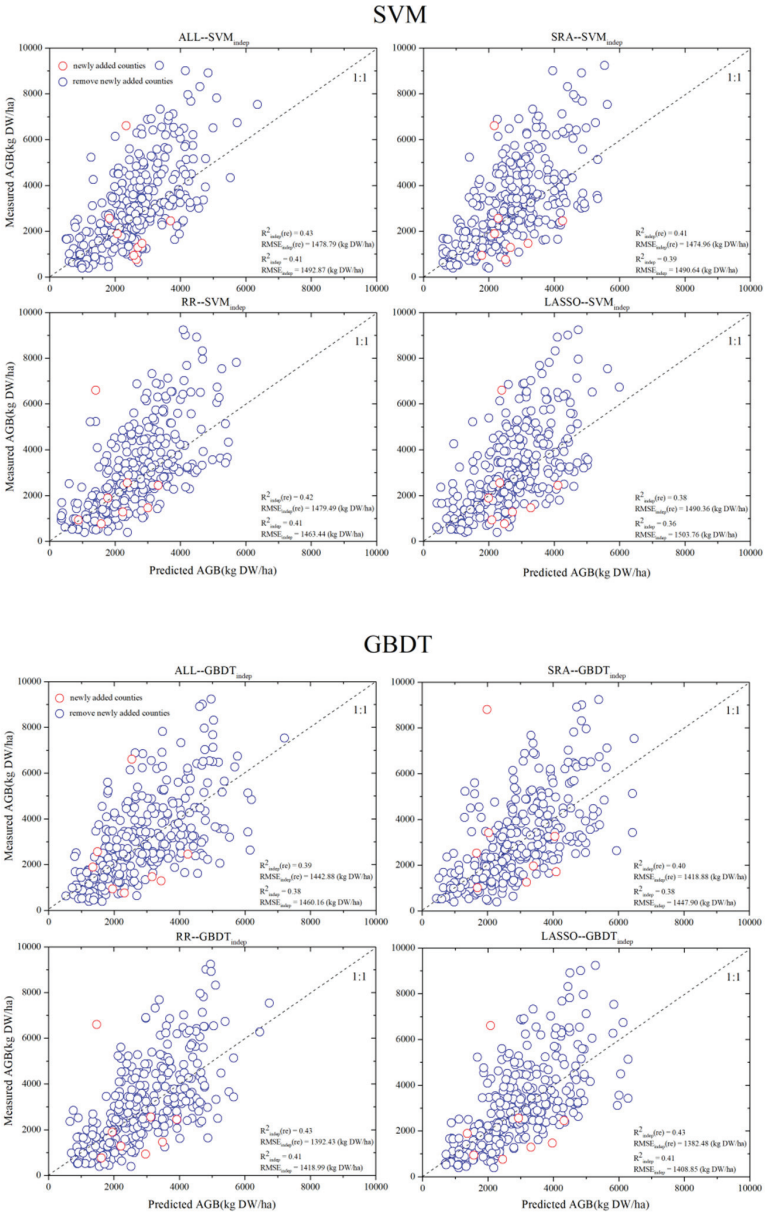


Figure 4. *Cont.*

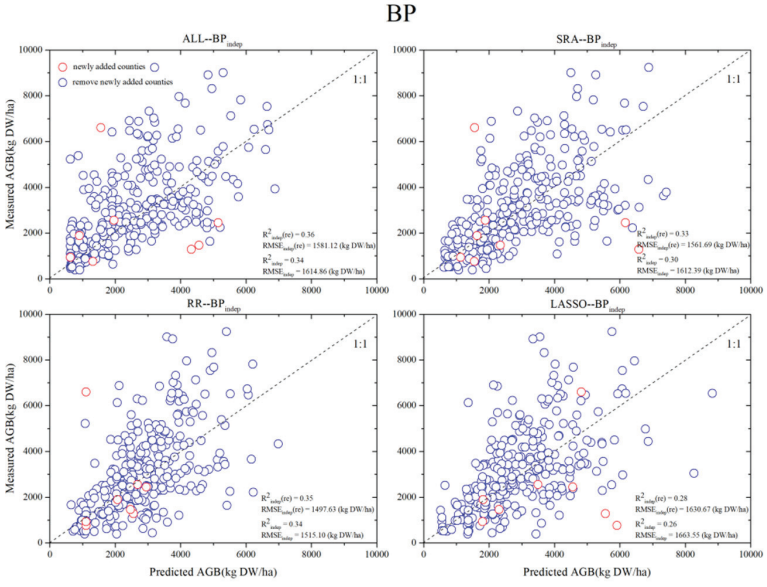


Figure 4. Assessing the accuracy of the independent AGB validation set simulated when expanding the spatio-temporal distribution by using different methods.

3.4. Spatial Distribution and Trend of Grassland Biomass Based on the RF-SRA Model

The RF-SRA model (the best model in this study) was used to simulate the annual maximum grassland aboveground biomass (the maximum aboveground biomass from July to September) in the study area for 11 years (2005 to 2015). Figure 5 illustrates the average results of the annual maximum grassland AGB in these 11 years (the average maximum AGB in the TRHR from 2005 to 2015 was 3267.41 ± 651.34 kg DW/ha). The results show that the higher grassland AGB was mainly concentrated in the eastern part of the study area and some of its western parts. For instance, Zaduo, Zhiduo, Qumalai, Maduo, and Northern Xinghai had lower distributions of grassland AGB. However, areas such as Xinghai, Guinan, and Guide also had lower altitudes and less AGB.

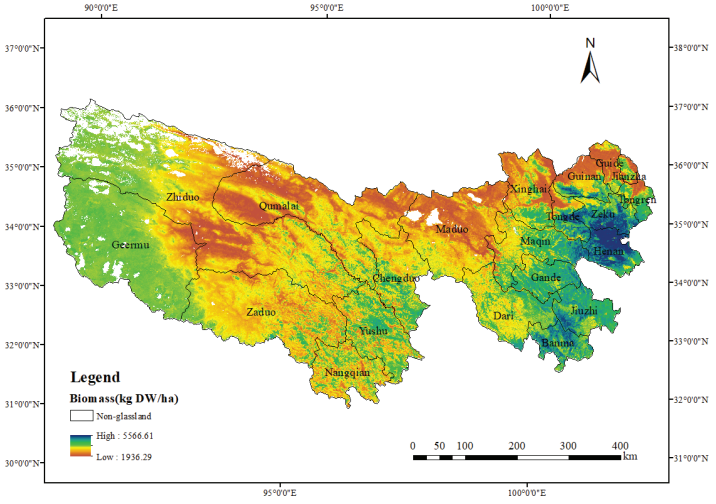


Figure 5. Distribution map of average grassland AGB in the TRHR from 2005 to 2015.

4. Discussion

4.1. AGB Mapping

In general, based on the inversion of the optimal model (RF-SRA), the annual spatial distribution of the largest grassland AGB showed an increasing trend from west to east and from north to south (Figure 5). This may be because the eastern region is a major pastoral area, with a higher temperature, a higher altitude, and a colder climate in the west of the TRHR (Figure 1). However, areas such as Xinghai, Guinan, and Guide have lower altitudes and less AGB, possibly due to a higher population density and more frequent human activities [32]. The generalized spatial distribution of AGB measurements during 2005–2015 provides an overall picture of the AGB values of the study area: an overall downward trend from southeast to northwest on the whole (Figure 2). This is consistent with the spatial variation of annual precipitation in the aboveground biomass of grassland [32]. At the same time, the annual average temperature gradually increased from west to east and was related to the trend of change in the longitude [33]. Precipitation and the annual mean temperature were positively correlated with grassland coverage in the Three-River Headwaters (Table S4). The spatial variation in grassland cover in this region may be influenced by both precipitation and annual temperature. The estimated spatial distribution map of AGB based on the RF models showed a reasonable spatial distribution, similar to that reflected in on-site measurements. A digital map can provide more details and cover a larger space than a limited field measurement (even though more than 1000 samples were collected).

4.2. Factors Affecting the Accuracy of the Remote Sensing Grassland AGB Estimation Model

Although the RF-SRA model attained accurate predictions, we think that its accuracy can be further improved. We analyze factors that affected the accuracy of the model:

(1) There were inevitable temporal differences between the biophysical parameters measured in the field and the satellite data during the peak growth period of the grasslands [34]. The field sampling time cannot be exactly the same as the time corresponding to the maximum vegetation index obtained from satellite data. In addition, the time period of this study was from 2005 to 2015. The first Sentinel-1 satellite was launched in 2014, so the Sentinel data of our study time are not available. TRHR is located in the hinterland of the Qinghai-Tibet Plateau. The high altitude and variable climate mean that it is often covered by clouds, which in turn leads to unusable Landsat data, that is, a lack of long-term continuous Landsat observation data. To obtain more variables and consider such practical difficulties as data availability within the study period, we selected MODIS data with a resolution of 500 × 500 m. However, in practice, the field sampling points are relatively small in number, and each pixel in the MODIS data covers an area of 500 × 500 m. Therefore, some differences were obtained in the spatial representation. In future work, more accurate and higher-resolution remote sensing data can be used, such as those obtained using unmanned aerial vehicles, to improve accuracy.

(2) Areas with complex terrain and slopes impacted reflectivity, which in turn affected the accuracy of the model. In addition, generally sparse grasslands (bare soil points) also affected some vegetation indices (such as the NDVI), which ultimately affected the model [35]. The grassland biomass measurements in this study were mainly distributed in the central and eastern regions of the TRHR. Grasslands in the western part of the TRHR are very sparse; many areas are deserts (Figure 1b). In addition, the western region has a higher altitude, a colder climate, and more complex terrain, which also introduced difficulties in sampling. We thus collected few and very concentrated samples in the western part of TRHR (only AGB data in the northeast of Geermu). This further affected the accuracy of the model.

(3) Uncertainty in field measurements also affected the model. For example, in-field measurements, the data collected are often affected by surface heterogeneity, human factors, and even traffic conditions. The data in this study cover a large span of time,

and there is a large amount of it. A time span that is too long and an amount of data that is too high can also lead to more errors in data measurement during the sorting process, which will inevitably affect the construction of the model.

4.3. Influence of the Number of Field Samples on the Model and the Model's Spatio-Temporal Scalability

The precision of AGB inversion models is highly dependent on the number of field samples. However, most studies have used fewer than 1000 field samples [17]. We measured continuous values of the grassland biomass in the TRHR for 11 years (1620 field samples) to explore the relationship between the field samples and the model. When constructing the AGB model, large differences were obtained in the structure and parameters of the model with the number of the field samples, and the accuracy of the simulation changed as well. Therefore, to better represent grasslands, more data points should be collected when sampling.

Previous studies have demonstrated that validation is key in this context. Without proper validation or a mechanistic understanding of the model, it is difficult to assess the quality of the results. Few studies have sought to estimate the validation error in AGB using ML [17]. AGB has traditionally been measured by destructive methods, which are limited to small areas due to their nature, time, expense, and the labor involved [36]. Therefore, evaluating the usefulness of the algorithm is important [37]. We used four criteria to evaluate the model. Figure 4 shows the results detailed in Table 3. Combining the graph and table comparison, it was found that model accuracy decreased when it was applied to the years without training data. When the model expands to an area with field sampling points that have not been incorporated into the model training, the model's accuracy will further decrease (Figure 4).

4.4. Input Variables to the Model

Environmental factors are important factors in determining the types, characteristics, and distribution of grasslands. Cui et al. (2015) found that the biomass of alpine grassland decreases with the increase of altitude. In this study, AGB showed a negative correlation with DEM (Table S4), which was the same as their findings [38]. However, the relationship between AGB and DEM in this paper is weak, which may be because the study did not set a certain altitude gradient when collecting points in the field. Moreover, when the samples were set, the research was mostly carried out on relatively flat grassland, which may also be the reason for the weak relationship between AGB and Aspect and Slope.

Soil is mainly composed of mineral particles, which can be divided into CL, SL, and SN according to their thickness. AGB was positively correlated with CL and negatively correlated with SN. Su et al. found that soils with higher CL usually have higher soil organic carbon, nutrient content and higher cation exchange capacity, and higher nutrient retention capacity and water holding effect to promote the growth of grassland vegetation [39]. The soil with higher SN has poor water holding effect, which is not conducive to the growth of vegetation. This is consistent with our results. AGB is negatively correlated with pH. The possible reason is that the pH of the study area is between 5.4 and 7.7. In the acidic soil, the species of microorganisms are limited and the decomposition of organic matter is slowed down, and the microbial activity is high in the neutral or alkaline environment [40], which is conducive to vegetation growth.

Among climatic factors, both AMP and AMT were positively correlated with AGB (Tables S4 and S5 and Figure S3, which may be because the increase of AMP and AMT promoted the growth of grassland vegetation. In the random forest importance ranking, GDD ranks second (Figure S4, and there is a negative correlation between GDD and AGB, which may be because the increase in GDD leads to faster plant development, but the actual growth season is shortened, resulting in a decrease in grassland AGB.

Satellite remote sensing is currently the most common and widely used regional-scale surface detection method. Satellite data can directly and timely capture biological growth

status through various spectral bands, and the products of various satellites have the same or complementary biological information, which is beneficial to grassland biomass prediction [5]. Different vegetation indices can reflect different biological characteristics of crops. For example, SAVI can indirectly reflect the canopy temperature of crops and reduce the influence of soil background on canopy reflectance [41]. In this study, OSAVI was the most important for the model (Figure S4. The OSAVI vegetation index is a modified SAVI, which differs from SAVI in that OSAVI takes into account the standard value of the canopy background adjustment factor (0.16). Therefore, when the canopy cover is low, this adjustment allows greater soil variation for OSAVI compared to SAVI. Therefore, higher predictability can be obtained.

5. Conclusions

Our study integrated 1620 measurement data on aboveground grassland biomass (AGB) with corresponding, continuously monitored remote sensing data from the GEE platform, meteorological data, topographic data, and soil characteristic data collected over 11 years in the TRHR of China. We then used the linear statistical method (PLS), ML methods (RF, SVM, and GBDT), and DL methods (BP) to establish grassland AGB estimation models. We then compared the models in terms of the accuracy of biomass predictions and simplicity. We also explored the spatio-temporal scalability of the linear regression model and the machine learning models. Overall, the ML models performed well. The RF models, based on the DEM, CL, pH, OR, OC, B1, B5, B6, OSAVI, D-LST, N-LST, and GDD, delivered the best performance. The estimated spatial distribution map of AGB based on the RF models was reasonably similar to the distribution of on-site measurements. It also provided more detail and covered a larger space than the limited field measurements do (even though more than 1000 samples were collected). This shows that when models are expanded in space and time, their accuracy decreases (as an example, the accuracy of the SRA-RF model decreased from 0.6 to 0.5). In future research, a process-based model that is derived from grassland AGB to train models could potentially be used to extend the spatio-temporal scalability of machine learning models. In addition, we also believe that ecosystem carbon sequestration is an interesting topic. In future work, we intend to explore whether the optimal model has the potential to be used in the development of emission factors for grassland areas from the perspective of addressing global climate change and combining the results of this study.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/rs14163843/s1, Figure S1: Schematic diagram of sampling, Figure S2: Flowchart illustrating the methodological steps undertaken to achieve the MODIS data processing in this study, Figure S3: Correlation between grassland AGB and precipitation, Figure S4: Input variable importance measure scatterplot, Table S1: Research site information, Table S2: Distribution of collection points in each county in the TRHR from 2005 to 2015, Table S3: Calculation of various indices of MODIS, Table S4: Correlation between grassland AGB and environmental factors, Table S5: Predicted AGB and precipitation, Text S1: The workings of Machine Learning algorithms.

Author Contributions: Formal analysis, J.G.; Investigation, Y.W., T.L., K.Z., N.C., W.Z. and F.Z.; Methodology, Y.W., R.Q., H.C., Q.F., M.H., J.L., C.L., Y.F. and J.H.; Project administration, F.Z.; Resources, F.Z.; Supervision, F.Z.; Validation, Y.W.; Writing—original draft, Y.W.; Writing—review & editing, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by [the Second Tibetan Plateau Scientific Expedition and Research] grant number [2019QZKK0305], [the National Natural Science Foundation of China] grant number [32071550] and [31770480], and [the '111' Programme] grant number [BP0719040].

Data Availability Statement: All data used in this manuscript are available upon reasonable request.

Acknowledgments: The authors are grateful to the High Performance Computing Center (HPCC) of Lanzhou University for performing the numerical calculations in this paper on its blade cluster system.

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

References

1. Yang, Q.; Liu, G.; Giannetti, B.F.; Agostinho, F.; MVBAlmeida, C.; Casazza, M. Emergy-based ecosystem services valuation and classification management applied to China's grasslands. *Ecosyst. Serv.* **2020**, *42*, 101073. [CrossRef]
2. Hensgen, F.; Bühle, L.; Wachendorf, M. The effect of harvest, mulching and low-dose fertilization of liquid digestate on above ground biomass yield and diversity of lower mountain semi-natural grasslands. *Agric. Ecosyst. Environ.* **2016**, *216*, 283–292. [CrossRef]
3. Zhou, W.; Li, H.; Xie, L.; Nie, X.; Wang, Z.; Du, Z.; Yue, T. Remote sensing inversion of grassland aboveground bio-mass based on high accuracy surface modeling. *Ecol. Indic.* **2021**, *121*, 107215. [CrossRef]
4. Wang, Z.B.; Ma, Y.K.; Zhang, Y.N.; Shang, J.L. Review of Remote Sensing Applications in Grassland Monitoring. *Remote Sens.* **2022**, *14*, 2903. [CrossRef]
5. Guan, K.; Wu, J.; Kimball, J.S.; Anderson, M.C.; Frolking, S.; Li, B.; Hain, C.R.; Lobell, D.B. The shared and unique values of optical, fluorescence, thermal and microwave satellite data for estimating large-scale crop yields. *Remote Sens. Environ.* **2017**, *199*, 333–349. [CrossRef]
6. Erica, G.; Andrew, H.; Rick, L. Using NDVI and EVI to Map Spatiotemporal Variation in the Biomass and Quality of Forage for Migratory Elk in the Greater Yellowstone Ecosystem. *Remote Sens.* **2016**, *8*, 404.
7. Gilabert, M.A.; González-Piqueras, J.; Garca-Haro, F.J.; Meliá, J. A generalized soil-adjusted vegetation index. *Remote Sens. Environ.* **2002**, *82*, 303–310. [CrossRef]
8. Ren, H.; Zhou, G.; Zhang, F. Using negative soil adjustment factor in soil-adjusted vegetation index (SAVI) for above-ground living biomass estimation in arid grasslands. *Remote Sens. Environ.* **2018**, *209*, 439–445. [CrossRef]
9. Qi, J.; Chehbouni, A.; Huete, A.R.; Kerr, Y.H.; Sorooshian, S. A modified soil adjusted vegetation index. *Remote Sens. Environ.* **1994**, *48*, 119–126. [CrossRef]
10. Guerschman, J.P.; Hill, M.J.; Renzullo, L.J.; Barrett, D.J.; Marks, A.S.; Botha, E.J. Estimating fractional cover of photosynthetic vegetation, non-photosynthetic vegetation and bare soil in the Australian tropical savanna region upscaling the EO-1 Hyperion and MODIS sensors. *Remote Sens. Environ.* **2009**, *113*, 928–945. [CrossRef]
11. Lobell, D.B.; Thau, D.; Seifert, C.; Engle, E.; Little, B. A scalable satellite-based crop yield mapper. *Remote Sens. Environ.* **2015**, *164*, 324–333. [CrossRef]
12. Nakano, T.; Bat-Oyun, T.; Shinoda, M. Responses of palatable plants to climate and grazing in semi-arid grasslands of Mongolia. *Glob. Ecol. Conserv.* **2020**, *24*, e01231. [CrossRef]
13. Wang, L.; Ali, A. Climate regulates the functional traits-aboveground biomass relationships at a community-level in forests: A global meta-analysis. *Sci. Total Environ.* **2020**, *761*, 143238. [CrossRef] [PubMed]
14. Verrelst, J.; Camps-Valls, G.; Munoz-Mari, J.; Rivera, J.P.; Veroustraete, F.; Clevers, J.; Moreno, J. Optical remote sensing and the retrieval of terrestrial vegetation bio-geophysical properties–A review. *Isprs J. Photo-Grammetry Remote Sens.* **2015**, *108*, 273–290. [CrossRef]
15. Tang, R.; Zhao, Y.T.; Lin, H.L. Spatio-Temporal Variation Characteristics of Aboveground Biomass in the Headwater of the Yellow River Based on Machine Learning. *Remote Sens.* **2021**, *13*, 3404. [CrossRef]
16. Xie, Y.; Sha, Z.; Yu, M.; Bai, Y.; Zhang, L. A comparison of two models with Landsat data for estimating above ground grassland biomass in Inner Mongolia, China. *Ecol. Model.* **2009**, *220*, 1810–1818. [CrossRef]
17. Morais, T.G.; Teixeira, R.F.M.; Figueiredo, M.; Domingos, T. The use of machine learning methods to estimate above-ground biomass of grasslands: A review. *Ecol. Indic.* **2021**, *130*, 108081. [CrossRef]
18. Craine, J.M.; Nippert, J.B.; Elmore, A.J.; Skibbe, A.M.; Hutchinson, S.L.; Brunsell, N.A. Timing of climate variability and grassland productivity. *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 3401–3405. [CrossRef]
19. Liu, S.; Cheng, F.; Dong, S.; Zhao, H.; Hou, X.; Wu, X. Spatiotemporal dynamics of grassland aboveground biomass on the Qinghai-Tibet Plateau based on validated MODIS NDVI. *Sci. Rep.* **2017**, *7*, 4182. [CrossRef]
20. Huete, A.; Didan, K.; Miura, T.; Rodriguez, E.; Gao, X.; Ferreira, L.G. Overview of the Radiometric and Biophysical Performance of the MODIS Vegetation Indices. *Remote Sens. Environ.* **2002**, *83*, 195–213. [CrossRef]
21. Zhu, X.; Liu, D. Improving forest aboveground biomass estimation using seasonal Landsat NDVI time-series. *Isprs J. Photogramm. Remote Sens.* **2015**, *102*, 222–231. [CrossRef]
22. Zhang, J.P.; Zhang, L.B.; Liu, W.L.; Qi, Y.; Wo, X. Livestock-carrying capacity and overgrazing status of alpine grass-land in the Three-River Headwaters region, China. *Geogr. Sci.* **2014**, *24*, 303–312. [CrossRef]
23. Hutchinson, M.F. *ANUSPLIN Version 4. 3 User Guide*; The Australia National University, Center for Re-source and Environment Studies: Canberra, Australia, 2004; Available online: http://cres.anu.edu.au/outputs/anusplin.php (accessed on 13 February 2021).

24. Chen, Y.; Shi, R.; Shu, S.; Gao, W. Ensemble and enhanced PM10 concentration forecast model based on stepwise regression and wavelet analysis. *Atmos. Environ.* **2013**, *74*, 346–359. [CrossRef]
25. Dorugade, A.V. New ridge parameters for ridge regression. *J. Assoc. Arab. Univ. Basic Appl. Sci.* **2014**, *15*, 94–99. [CrossRef]
26. Zhang, Y.; Ma, F.; Wang, Y. Forecasting crude oil prices with a large set of predictors: Can LASSO select powerful predictors? *J. Empir. Financ.* **2019**, *54*, 97–117. [CrossRef]
27. Metz, M.; Abdelghafour, F.; Roger, J.; Lesnoff, M. A novel robust PLS regression method inspired from boosting prin-ciples: RoBoost-PLSR. *Anal. Chim. Acta* **2021**, *1179*, 338823. [CrossRef]
28. Wang, Y.; Wu, G.; Deng, L.; Tang, Z.; Wang, K.; Sun, W.; Shangguan, Z. Prediction of aboveground grassland biomass on the Loess Plateau, China, using a random forest algorithm. *Sci. Rep.* **2017**, *7*, 6940. [CrossRef]
29. Li, W.; Yan, X.; Pan, J.; Liu, S.; Xue, D.; Qu, H. Rapid analysis of the Tanreqing injection by near-infrared spectroscopy combined with least squares support vector machine and Gaussian process modeling techniques. *Spectrochim. Acta Part A Mol. Biomol. Spectrosc.* **2019**, *218*, 271–280. [CrossRef]
30. Friedman, J.H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367–378. [CrossRef]
31. Wang, K.; Ho, C.; Tian, C.; Zong, Y. Optical health analysis of visual comfort for bright screen display based on back propagation neural network. *Comput. Methods Programs Biomed.* **2020**, *196*, 105600. [CrossRef]
32. Yang, H.; Xiao, H.; Guo, C.; Sun, Y. Spatial-temporal analysis of precipitation variability in Qinghai Province, China. *Atmos. Res.* **2019**, *228*, 242–260. [CrossRef]
33. Jin, H.J.; Luo, D.L.; Wang, S.L.; Lv, L.Z.; Wu, J.C. Spatiotemporal variability of permafrost degradation on the Qinghai-Tibet Plateau. *Sci. Cold Arid. Reg.* **2011**, *3*, 281–305.
34. Yuan, X.L.; Tian, L.H.; Luo, G.P.; Chen, X. Estimation of above-ground biomass using MODIS satellite imagery of multiple land-cover types in China. *Remote Sens. Lett.* **2016**, *7*, 1141–1149. [CrossRef]
35. Yang, S.; Feng, Q.; Liang, T.; Liu, B.; Zhang, W.; Xie, H. Modeling grassland above-ground biomass based on artificial neural network and remote sensing in the Three-River Headwaters Region. *Remote Sens. Environ.* **2018**, *204*, 448–455. [CrossRef]
36. Catchpole, W.R.; Wheeler, C.J. Estimating plant biomass: A review of techniques. *Aust. J. Ecol.* **1992**, *17*, 121–131. [CrossRef]
37. Wu, H.; Li, Z. Scale Issues in Remote Sensing: A Review on Analysis, Processing and Modeling. *Sensors* **2009**, *9*, 1768–1793. [CrossRef]
38. Cui, H.J.; Wang, G.X.; Yang, Y.; Yang, Y. Variation of Quantitative Characteristics of Alpine Grassland Plant Community along the Altitude Gradient and Its Influencing Factors. *J. Ecol.* **2015**, *34*, 3016–3023. (In Chinese)
39. Su, Y.Z.; Wang, J.Q.; Yang, R.; Yang, X. Soil texture controls vegetation biomass and organic carbon storage in arid desert grassland in the middle of Hexi Corridor region in Northwest China. *Soil Res.* **2015**, *53*, 366–376. [CrossRef]
40. Li, Z.; Sun, B.; Lin, X.X. The Density of Soil Organic Carbon and the Controlling Factors of Its Transformation in Eastern China. *Geogr. Sci.* **2001**, *04*, 301–307. (In Chinese)
41. Carpintero, E.; Mateos, L.; Andreu, A.; González-Dugo, M.P. Effect of the differences in spectral response of Mediterranean tree canopies on the estimation of evapotranspiration using vegetation index-based crop coefficients. *Agric. Water Manag.* **2020**, *238*, 106–201. [CrossRef]

applied sciences

MDPI

Article

Recursive Least Squares for Near-Lossless Hyperspectral Data Compression

Tie Zheng [1,2], Yuqi Dai [1,2], Changbin Xue [1,*] and Li Zhou [1]

[1] National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China; zhengtie17@mails.ucas.ac.cn (T.Z.); daiyuqi18@mails.ucas.ac.cn (Y.D.); zhouli@nssc.ac.cn (L.Z.)
[2] Department of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: xuechangbin@nssc.ac.cn

Abstract: The hyperspectral image compression scheme is a trade-off between the limited hardware resources of the on-board platform and the ever-growing resolution of the optical instruments. Predictive coding attracts researchers due to its low computational complexity and moderate memory requirements. We propose a near-lossless prediction-based compression scheme that removes spatial and spectral redundant information, thereby significantly reducing the size of hyperspectral images. This scheme predicts the target pixel's value via a linear combination of previous pixels. The weight matrix of the predictor is iteratively updated using a recursive least squares filter with a loop quantizer. The optimal number of bands for prediction was analyzed experimentally. The results indicate that the proposed scheme outperforms state-of-the-art compression methods in terms of the compression ratio and quality retrieval.

Keywords: near-lossless compression; recursive least squares; hyperspectral image; predictive coding

Citation: Zheng, T.; Dai, Y.; Xue, C.; Zhou, L. Recursive Least Squares for Near-Lossless Hyperspectral Data Compression. *Appl. Sci.* **2022**, *12*, 7172. https://doi.org/10.3390/app12147172

Academic Editors: Yue Wu, Kai Qin, Qiguang Miao and Maoguo Gong

Received: 22 June 2022
Accepted: 13 July 2022
Published: 16 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Hyperspectral images from space-borne spectrometers play a crucial role in multifarious aspects, including geological exploration, environmental monitoring, and material identification [1]. Researchers continue to enhance the spectral and spatial resolutions of the instruments, and the size of a hyperspectral image is currently more than hundreds of megabytes (MBs) [2–4]. Nevertheless, such a wealth of information places excessive demands on the transmission and storage processes. Data compression has proven to be an effective way to alleviate this issue [5,6].

The compression techniques for hyperspectral images are divided into three categories: lossless, lossy, and near-lossless compression [7]. Lossless compression allows for reconstruction of the original image, ideally at the price of a limited compression ratio. Lossy compression approximates the original image while generally minimizing distortion in the l_2-norm. It tolerates a small amount of information distortion between the original image I and the reconstruction image $\hat{I}$, which allows for a high compression ratio. Near-lossless compression aims to achieve a higher compression ratio than lossless techniques by allowing pixel-level distortion. It strictly bounds the l_∞-norm by setting the peak absolute error (PAE) [8]. The user-specified parameter guarantees the max. absolute distortion so that $PAE \leq \Lambda$ defines a limited error range for reconstruction of individual pixels. As a result, employing a proper parameter Λ makes the compression process almost lossless. It is a well-known fact that the quality of images is affected by the inherent noise of the device [9,10]. When the maximum error introduced by the compression process is smaller than the background noise, the quality of the reconstructed image is almost similar to that obtained with lossless compression.

Details on near-lossless compression techniques are discussed in a later section. Most near-lossless compression techniques can be roughly classified into three categories:

prediction-based subsequent quantization coding, lossless coding based on pre-quantization, and two-stage near-lossless encoding [11].

Predictive-based coding, one of the most popular schemes, enables low-complexity, high-throughput solutions [12–14]. These schemes first compute the prediction value of the target pixel from the previous encoding. The difference between the predicted and the original pixel value is known as the prediction error. Subsequently, a near-lossless compression scheme is obtained by encoding the quantized prediction error. The two most typical prediction schemes are JPEG-LS [15] and CALIC [16], which are widely used to process two-dimensional images. Meanwhile, the CALIC algorithm, with its better compression performance, is extended to 3D-CALIC [17] and M-CALIC [18] based on the correlation of hyperspectral images. However, the CALIC-based extension scheme cannot effectively remove the redundancy of hyperspectral images and is not friendly to hardware implementation. The Consultative Committee for Space Data Systems (CCSDS) proposes the standard CCSDS-123, which is based on the signed least mean square (SLMS) filter [19,20]. This compression scheme has low complexity and excellent compression results. NL-CCSDS-123 [10] and CCSDS-123-AC [21] are two near-lossless extended versions that both rely on the predictor of CCSDS-123. The NL-CCSDS-123 scheme encodes the quantized residual pixels after using a range coder, whereas CCSDS-123-AC employs a lightweight context-based arithmetic encoder. However, since their predictor uses a simple function to update the weight coefficients, the prediction accuracy can be further improved.

The second type of near-lossless compression category is based on pre-quantizing the original pixels with a quantizer and then applying a lossless compression technique. It is widely known that such a model is suboptimal, and the compression results are not outstanding [22]. However, it is suitable for a scene with high-speed compression demands since it does not need to include a feedback loop. S.-C Tai et al. proposed the Pre-CCSDS-IDC [23] compression scheme in order to improve the compression rate without modifying the existing CCSDS-IDC hardware system [24]. It can import the pre-quantized images into CCSDS-IDC directly. In [25] a pre-quantization compression scheme is implemented based on ground-based CNN reconstruction. The spaceborne part can be considered a combination of the pre-quantizer and CCSDS-123 predictor. Eventually, the CNN is employed as a feature extractor on the ground to do a secondary reconstruction of the decoded image, which leads to a higher signal-to-noise ratio.

The third near-lossless compression category builds on a combination of lossy and lossless compression. First, the reconstructed images are obtained by lossy compression. Then, the differences between these and the original image are quantized and encoded. X. Wu proposes an approximate lossless image compression scheme that combines wavelets and CALIC [26]. It uses CALIC to compress the residual image between the wavelet approximation and the original image. C.-W Chen employed CCSDS-IDC for the lossy phase, followed by bit-plane encoding (BPE) coding. However, the scheme does not obtain the optimal lossy bit rate. J.Beerten combines JPEG2000, as a lossy layer, with a near-lossless layer consisting of BPE and arithmetic coding [27]. This scheme uses computationally expensive iterative methods to determine the optimal lossy bitrate and obtain a competitive coding performance.

Due to the characteristics of the prediction-based compression scheme, it gradually replaces the transform-based algorithm in the on-board compression platform [28]. In this paper, we focus on the prediction-based quantization technique, which aims to enhance the performance of the near-lossless compression scheme by fully exploiting the spatial–spectral redundancy of hyperspectral images. The proposed method combines recursive least squares (RLS) with an in-loop quantizer and, subsequently, encodes the quantization residuals using an entropy encoder. Ultimately, a competitive compression ratio is produced while guaranteeing the quality of the reconstructed image.

The rest of the paper is organized as follows: Section 2 provides a detailed description of the near-lossless feedback loop's compression framework with the RLS prediction.

Section 3 presents the experimental dataset and comparative results of this method with an analysis and discussion. Finally, Section 4 concludes this paper.

2. Compression Scheme

In this work, a loop predictor is used for near-accurate prediction of target pixels by removing the correlation of hyperspectral images. The input image data is passed through the loop predictor point by point. The differences between the predicted and original values are quantized. Subsequently, the entropy encoder encodes the quantized residual image and outputs the compressed codestream. The same algorithm with the same parameters must be used at the decoding stage during the reconstruction process after inverse entropy coding. Figure 1 presents a pictorial representation of the compression scheme, which explains the essential steps. The five significant steps of the proposed scheme, including spatial predictor, spectral predictor, controlled quantization, sample representative, and entropy coding, are discussed in the following subsections.

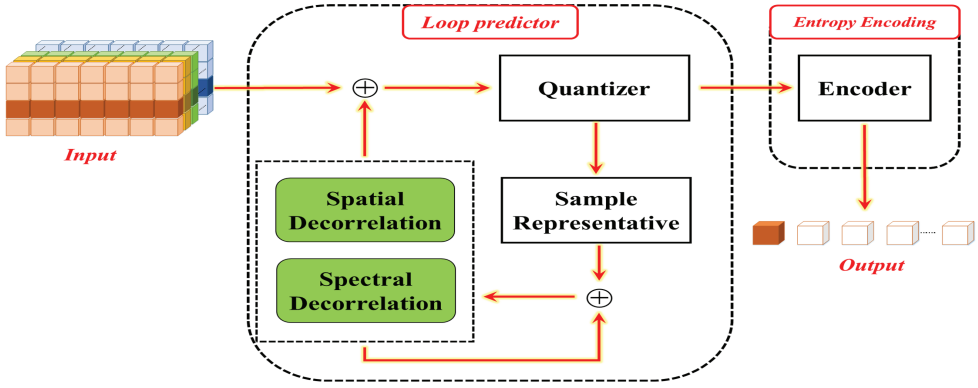


Figure 1. Compressor schematic.

2.1. Loop Predictor

Hyperspectral images can be regarded as a three-dimensional data. Let $s_z(x, y)$ denote the value of the original pixel in the zth band at the xth row with the yth column. We also use $s_z(n)$ to represent the nth pixel in band z, where the notation n means the pixel is derived from $n = NX * y + x$ in the spatial domain. NX, NY, and NZ provide the image's width, height, and number of bands.

The spatial prediction estimate is computed using a causal, linear predictor for each pixel. To remove spatial correlation from the hyperspectral image, the preliminary estimate $\hat{S}_z(n)$of the target pixel $S_z(n)$ is generated by averaging the pixels in the context window of the same band. The context window of the target pixel can be explained using Figure 2. The parameter r denotes the local context window's radius, and the target pixel points are represented in orange. The blue and green parts show the neighborhood windows with radii of one and three, respectively.

Then, the spatial prediction error $d_z(n)$ for the nth pixel point in the zth band is defined as Equation (1).

$$d_z(n) = s_z(n) - \hat{S}_z(n). \tag{1}$$

Note that in order to ensure the feasibility of the decompressor, the preliminary prediction error $d_z(n)$ of the first spectrum is directly encoded with the entropy encoder that follows. The others are encoded after removing the inter-spectral redundancy.

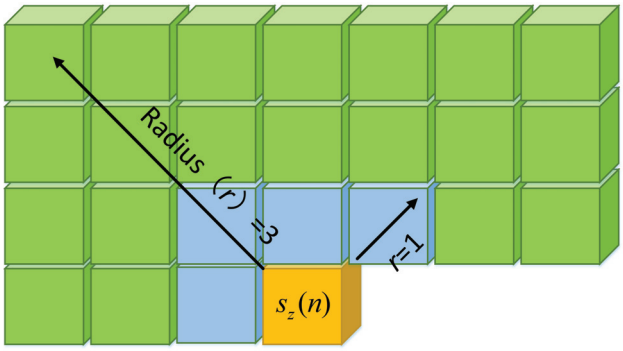


Figure 2. The context window of current pixel.

In the spectrum prediction process, the RLS filter input vector is formed by $\boldsymbol{d}_{z,k}(n) = [d_{z-k}(n), \cdots, d_{z-1}(n)]^T$, where k is the number of history bands used to predict the current pixel, i.e., the prediction length. The corresponding weight vector is $\boldsymbol{w}_{z,k}(n) = [w_{z-k}(n), \cdots, w_{z-1}(n)]^T$. Then, the RLS predictor is initialized as Equation (2).

$$\boldsymbol{k}_z(0) = 0, \boldsymbol{w}_z(0) = 0, \boldsymbol{p}_z(0) = 0. \tag{2}$$

The spectrum prediction residual $e_z(n)$ is calculated as Equation (3).

$$e_z(n) = d_z(n) - \boldsymbol{d}_{z,k}(n)\boldsymbol{w}_{z,k}^T(n-1). \tag{3}$$

The parameter's peak absolute error Λ determines the maximum allowable absolute difference between the original and reconstructed pixel values. Each predicted residual corresponds to a quantized residual $q_z(n)$ defined by Equation (4).

$$q_z(n) = sgn(e_z(n)) \times \left\lfloor \frac{\Lambda + |e_z(n)|}{2\Lambda + 1} \right\rfloor, \tag{4}$$

where the $sgn(x)$ is a function that extracts the sign value of x. For spectral images that need to be stored with absolute accuracy, a lossless compression mode can be used by setting $\Lambda = 0$. That is, $q_z(n) = e_z(n)$, thus ensuring that the decoding process can accurately reconstruct the original sample. However, when using a non-zero peak absolute error, $q_z(n)$ represents an approximation of the above prediction error, rather than the actual value.

Since the image introduces distortion affected by the quantizer, the compressed code stream cannot be directly employed to reconstruct the sample. In order to guarantee synchronization with the decompression stage, the reconstruction value $\widetilde{s}_z(n)$ should be calculated for each pixel, as shown in Equation (5). Then, $\widetilde{s}_z(n)$ is used in the prediction process for the next pixel point to be measured.

$$\widetilde{s}_z(n) = \hat{s}_z(n) + \boldsymbol{d}_{z,k}(n)\boldsymbol{w}_{z,k}(n-1) + \widetilde{e}_z(n). \tag{5}$$

where the center of the predicted residual reconstruction value $\widetilde{e}_z$ is calculated by Equation (6).

$$\widetilde{e}_z(n) = q_z(n)(2\Lambda + 1), \tag{6}$$

The gain of RLS is

$$\boldsymbol{k}_z^T(n) = \frac{\boldsymbol{p}_z(n-1)\boldsymbol{d}^T{}_z(n)}{1 + \boldsymbol{d}_z(n)\boldsymbol{p}(n-1)_z\boldsymbol{d}^T{}_z(n)}, \tag{7}$$

$$\boldsymbol{p}_z(n) = \boldsymbol{p}_z(n-1) - \boldsymbol{k}_z^T(n)\boldsymbol{d}_z(n)\boldsymbol{p}_z(n-1), \tag{8}$$

where $\boldsymbol{p}(n)$ is an auxiliary vector required to reduce the computational burden.

Then, the weight vector $\boldsymbol{w}_{z,k}(n)$ is updated by the recursive Equation 9.

$$\boldsymbol{w}_{z,k}(n) = \boldsymbol{w}_{z,k}(n-1) + \boldsymbol{k}_z(n)\widetilde{e}_z(n). \tag{9}$$

Finally, the predictor is executed for each pixel in raster scan order until the last pixel is reached.

2.2. Entropy Encoding

Entropy coding techniques are used to encode the residual error after quantization in predictive compression. The adaptive arithmetic encoder is adopted in the encoding stage, whose compression ratio is near the theoretical entropy. The quantization residual is represented by 16 bits, meaning that the arithmetic code's codebook needs 65,536 symbols. Based on the probability distribution of the prediction residual, most symbols are not used. Therefore, an adaptive codebook is adopted. The initial codebook contains two symbols: 0 and ESC. When a new symbol needs to be encoded, the encoder will use the probability of ESC to encode it, and the 16-bit symbol will be appended to the code stream. The code book absorbs the new symbol after the symbol has been output to the stream.

3. Experimental Results

The proposed scheme, named near-lossless recursive least squares (NLRLS), has two parameters, namely, the radius of the context window r and the prediction length k. Since these two parameters critically impact the compression results, the selection of optimal values is explained in the first subsection. Furthermore, the compression performance of the proposal is compared with several state-of-the-art schemes, including compression results in the distortion metrics of the reconstructed image. This section highlights the corresponding results and analysis.

The platform for testing is a personal computer powered by a single Intel Core i7-7700K central processing unit (CPU) at 4.2GHz with 16GB random access memory (RAM). We adopted the standard hyperspectral image test data recommended by the International Consultative Committee for Space Data Systems, which includes Atmospheric Infrared Sounder (AIRS) and Airborne Visible/Infrared Imaging Spectrometer Calibrated and Uncalibrated (AC and AU) [29]. Each of them has a bit depth of 16 bit-per-pixel (BPP). Table 1 details the dataset used in the tests, including sensor abbreviations, scene names, and dimensions.

Table 1. The sensor names and their main features.

Sensor	Scene	Number of Scence	Rows	Columns	Bands	Formation
AC	Yellowstone	5	677	512	224	Signed 16 bit
AU	Yellowstone	3	680	512	224	Unsigned 16 bit
AIRS	Gran	8	90	135	1501	Unsigned 16 bit

3.1. Parameter Settings

The accuracy of the RLS predictor has a strong correlation with the prediction length k of the input vector $\boldsymbol{d}_{z,k}(n)$. Additionally, to preserve the causality of the pixels in the context window, the local mean predictor is used to estimate $\hat{S}_z(n)$. Therefore, the radius r of the context window is another important parameter that directly affects the compression results. In order to evaluate the effects of the k and r parameters on near-lossless compression performances, average bit rates of the proposed scheme at different peak absolute errors are shared in Figure 3 for the AC datasets.

It can be seen from Figure 3a–d that the proposed compression schemes exhibit similar radio-compressed bit-rate characteristics for different peak absolute errors. The lowest bit rates are observed in the case of low r–high k parameter pairs. If the radius is treated as

a constant, we can see that the deceleration of the compressed bit rate gradually slows down as the prediction length increases. Moreover, the RLS predictor update requires the computation of $\boldsymbol{P}(n)$ as well as $\boldsymbol{d}_{z,k}(n)$. The algorithm has an $O(k^2)$ computational complexity, meaning that the computing resource consumption increases quadratically with k. Considering the dual effects of the actual compression results and the computational complexity, the initial radio r and prediction length k parameters are selected as 2 and 12, respectively.

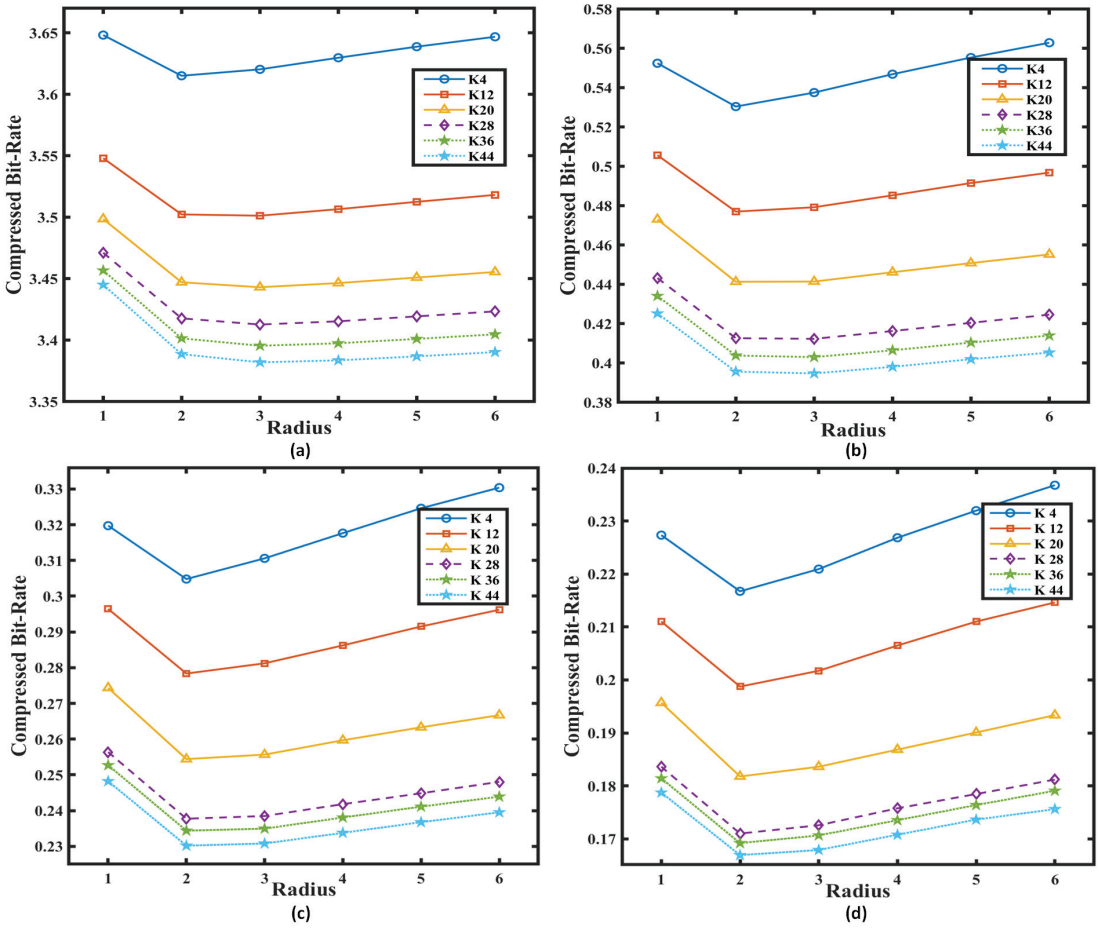


Figure 3. Average compressed bit-rate for different peak absolute errors: (**a**) encoded at $\Lambda = 0$, (**b**) encoded at $\Lambda = 10$, (**c**) encoded at $\Lambda = 20$, (**d**) encoded at $\Lambda = 30$.

3.2. Compression Performance Analysis

3.2.1. Compression Results

A total of 16 scene data from three types of hyperspectral image datasets (AC, AU, AIRS) were used as test data sources. The results of CCSDS-123-AC, NLCCSDS-123, and M-CALIC are listed here to compare with the proposed near-lossless compression scheme. Table 2 shows the average compression results for multiple compression schemes at five different peak absolute errors.

Table 2 reports compression results in terms of the bit rate. The second column reports the first-order entropy on average (in bit-per-pixel) for each scene type. It represents the

entropy of individual pixels, regardless of any correlation among the pixels. In addition, each subsequent column shows the average compression results for a compression scheme at different peak absolute errors Λ. The best compression results for each test are indicated in bold. The bracketed text indicates the coding gain of our method relative to other techniques, and a positive difference means our scheme is better.

Table 2. Comparison of different near-lossless schemes for several different peak absolute errors.

Sensor Abbreviation	Entropy	Λ Values	CCSDS-123-AC	NL-CCSDS-123	M-CALIC	NLRLS
AC	9.77	$\Lambda = 0$	3.66 (0.14)	3.73 (0.21)	4.03 (0.51)	**3.52**
		$\Lambda = 1$	2.45 (0.38)	2.54 (0.47)	2.87 (0.80)	**2.07**
		$\Lambda = 10$	0.58 (0.10)	0.94 (0.46)	0.88 (0.40)	**0.48**
		$\Lambda = 20$	0.35 (0.07)	0.72 (0.44)	0.53 (0.25)	**0.28**
		$\Lambda = 30$	0.26 (0.06)	0.63 (0.43)	0.40 (0.20)	**0.20**
AU	12.13	$\Lambda = 0$	5.87 (0.14)	5.95 (0.22)	6.13 (0.40)	**5.73**
		$\Lambda = 1$	4.80 (0.66)	4.89 (0.75)	5.05 (0.91)	**4.14**
		$\Lambda = 10$	1.96 (0.38)	2.24 (0.66)	2.22 (0.64)	**1.58**
		$\Lambda = 20$	1.14 (0.22)	1.46 (0.54)	1.40 (0.48)	**0.92**
		$\Lambda = 30$	0.83 (0.17)	1.18 (0.52)	1.05 (0.39)	**0.66**
AIRS	11.39	$\Lambda = 0$	4.25 (0.41)	4.31 (0.47)	4.38 (0.54)	**3.84**
		$\Lambda = 1$	3.08 (0.77)	3.13 (0.82)	3.21 (0.90)	**2.31**
		$\Lambda = 10$	0.61 (0.28)	0.98 (0.65)	0.70 (0.37)	**0.33**
		$\Lambda = 20$	0.29 (0.15)	0.66 (0.52)	0.36 (0.22)	**0.14**
		$\Lambda = 30$	0.20 (0.10)	0.57 (0.47)	0.26 (0.16)	**0.10**

It can be seen that the compression bit rate of each technique decreases rapidly as a function of Λ. The reported results indicate that our scheme outperforms the other known schemes for all sensors. For lossless coding ($\Lambda = 0$), our scheme beats the other schemes by a slight margin. For near-lossless encoding ($\Lambda > 0$), our scheme provides outstanding compression results; i.e., it clearly yields the lowest ratio of all the compared methods. Compared to CCSDS-123-AC, which has the best-known compression results, on average, the proposal provides benefits ranging from 0.11 to 0.6 bpp, depending on the allowed absolute error of the peak.

The previous subsection stated that the computational complexity of NLRLS is $O(k^2)$. The computational complexity of the SLMS filter used by CCSDS-123-AC is $O(k)$, where k is the number of bands. Although NLRLS has higher complexity than CCSDS-123-AC, it produces better convergence and more accurate predictions. Therefore, the compression results of the NLRLS scheme are better.

Graphically, the relative gain of our scheme over CCSDS-123-AC is represented in Figure 4 for different datasets. Each color represents the comparison results using different datasets, and each cluster indicates a different peak absolute error. Therefore, the higher percentage means that our scheme is superior. The results indicate that the proposed scheme provides an average optimization from 5.29% to 31.19% in bit-per-pixel change for different peak absolute errors. The AIRS images, especially, outperform by almost 50% with $\Lambda \geq 10$.

We note that complex image preprocessing can achieve better compression results, such as band reorder, clustering, and super-pixel methods [30,31]. However, this is beyond the scope of this paper and will not be discussed here.

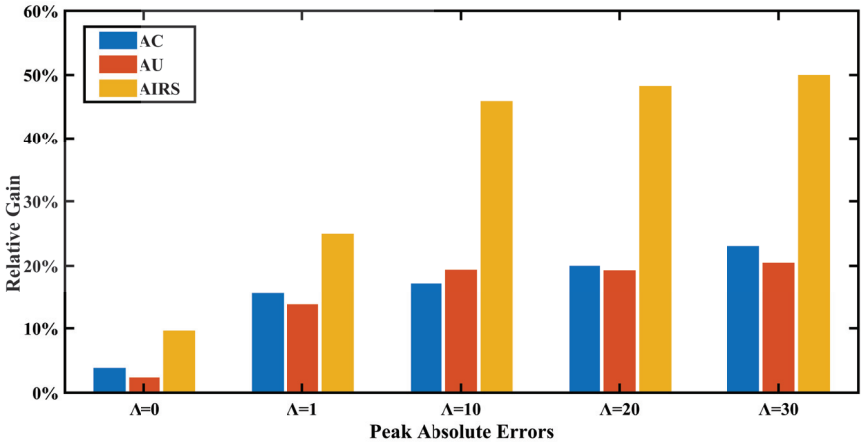


Figure 4. The relative gain of our scheme on CCSDS-123-AC at different peak absolute errors for AC, AU, and AIRS datasets.

3.2.2. Reconstruction Quality Analysis

In the near-lossless compression scheme, the peak absolute error Λ introduced by the compression process directly affects the quality of the reconstructed image. Figure 5 shows the original image of the AIRS Grand16 image and the reconstructed image at $\Lambda = \{0, 1, 10, 20, 30\}$. It can be noted that the visual performance of each image is extremely similar.

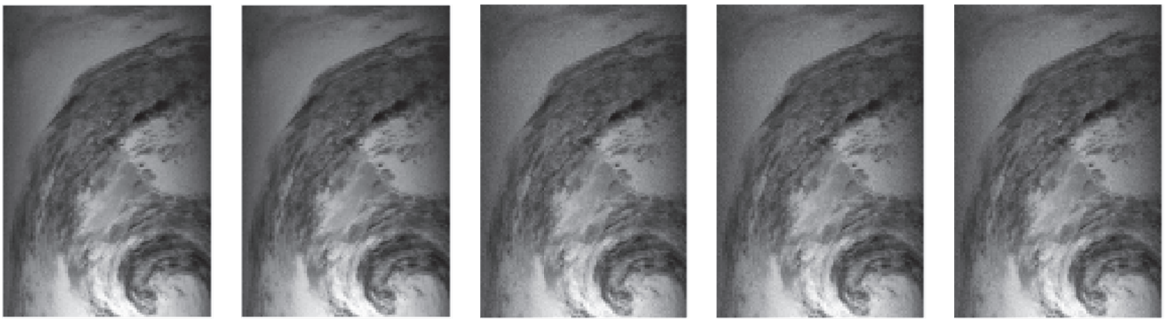


Figure 5. Visual comparison for the "AIRS Grand16 1256th band" image: (**a**) original, (**b**–**e**) reconstructed image at $\Lambda = \{1, 10, 20, 30\}$, respectively.

To objectively evaluate the quality of the reconstructed images, we compare the proposed scheme with the best-known near-lossless compression scheme CCSDS-123-AC, including two metrics, namely the peak signal-to-noise ratio and spectral angle mapper.

If D is the dynamic range (in bits) of the original image, the maximum pixel value is $2^D - 1$. The reconstructed image quality is evaluated in terms of the peak signal-to-noise ratio (PSNR), which is measured in dB and defined as:

$$PSNR\left(S, \widetilde{S}\right) = 10 \log_{10} \frac{\left(2^D - 1\right)^2}{MSE\left(S, \widetilde{S}\right)} (dB)\,, \tag{10}$$

where $MSE\left(S,\widetilde{S}\right)$ is the mean square error between the original image S and the reconstructed image $\widetilde{S}$.

$$MSE\left(S,\widetilde{S}\right)=\frac{\sum_1^{NX}\sum_1^{NY}\sum_1^{NZ}\left(S_z(x,y)-\widetilde{S}_z(x,y)\right)^2}{NX\times NY\times NZ}, \tag{11}$$

Spectral Angle Mapper (SAM) treats the spectrum of each image element as a high-dimensional vector and measures the similarity between two spectra by calculating the angle of the vector. The SAM of each pixel spectrum in the original and reconstructed image is denoted as

$$\alpha(x,y)=\cos^{-1}\left(\frac{\sum_Z^{NZ}S_z(x,y)\times\widetilde{S}_z(x,y)}{\sqrt{\sum_Z^{NZ}S_z^2(x,y)}\sqrt{\sum_Z^{NZ}\widetilde{S}_z^2(x,y)}}\right), \tag{12}$$

The average spectral angle is used to calculate the spectral variability of the reconstructed image, and, the smaller the angle, the less distortion.

We experimentally obtained a series of reconstructed image data by varying the maximum allowable peak absolute error. Figures 6 and 7 show the PSNR and the average SAM variation with the bit rate, respectively.

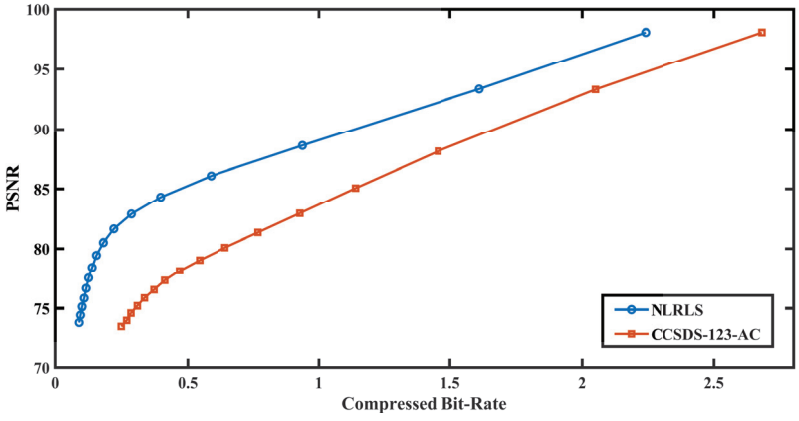


Figure 6. The PSNR performance of the proposed method compared to CCSDS-123-AC.

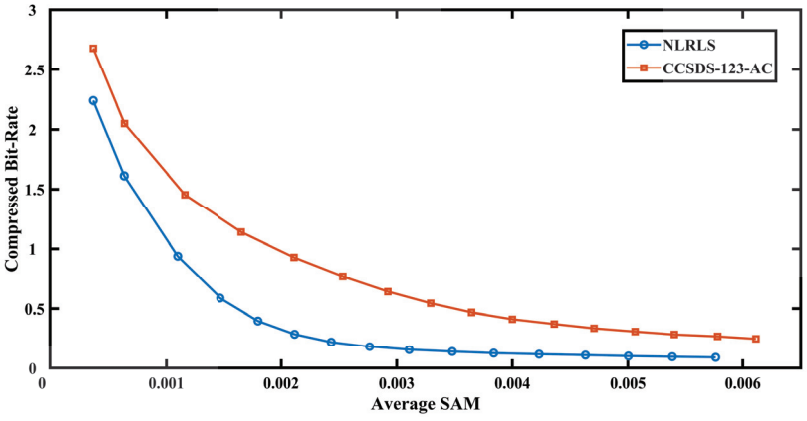


Figure 7. The SAM performance of the proposed method compared to CCSDS-123-AC.

The experimentally obtained data points have been highlighted by marked dots, and linear interpolation is used between the markers. It can be seen that under the same compressed bit rate, the proposed scheme has a higher peak signal-to-noise ratio and a smaller average spectral angle. Further, for image quality, as measured by the PSNR and SAM, our approach is competitive compared to the best-known CCSDS-123-AC scheme.

4. Conclusions

In this paper, the prediction-based near-lossless compression technique is used to reduce the size of the hyperspectral image. The target pixel is predicted from the combination of previous pixels in the spatial and spectral bands. The coefficients are predicted using the weight matrix of the RLS filter with an in-loop quantizer. The experiments were performed on three types of CCSDS hyperspectral images' datasets, including three, five, and eight scenes, respectively. The optimal number of bands for the loop predictor was analyzed experimentally. The results indicate that the proposed scheme provides an average optimization from 5.29% to 31.19% in bit-per-pixel for different peak absolute errors and achieves a competitive reconstructed image quality compared to the state-of-the-art methods.

In the future, we plan to develop an automated model for parallel processing of near-lossless compression schemes and feasible hardware implementation solutions for space-based platforms. This research work will further pave the way for future developments in the field of deep space exploration.

Author Contributions: Conceptualization, T.Z.; methodology, T.Z. and Y.D.; investigation, T.Z. and C.X.; formal analysis, C.X., L.Z. and T.Z.; writing—original draft, T.Z.; writing—review and editing, Y.D. and C.X.; funding acquisition, C.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Chinese Academy of Sciences Project, grant number: CXJJ-20S017.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Luo, J.; Wu, J.; Zhao, S.; Wang, L.; Xu, T. Lossless compression for hyperspectral image using deep recurrent neural networks. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 2619–2629. [CrossRef]
2. Santos, L.; Gomez, A.; Sarmiento, R. Implementation of CCSDS standards for lossless multispectral and hyperspectral satellite image compression. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *56*, 1120–1138. [CrossRef]
3. Bascones, D.; Gonzalez, C.; Mozos, D. A Real-Time FPGA Implementation of the CCSDS 123.0-B-2 Standard. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5525113. [CrossRef]
4. Fjeldtvedt, J.; Orlandic, M.; Johansen, T.A. An Efficient Real-Time FPGA Implementation of the CCSDS-123 Compression Standard for Hyperspectral Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3841–3852. [CrossRef]
5. Song, J.; Zhou, L.; Deng, C.; An, J. Lossless compression of hyperspectral imagery using a fast adaptive-length-prediction RLS filter. *Remote Sens. Lett.* **2019**, *10*, 401–410. [CrossRef]
6. Zheng, T.; Xue, C.; Song, J. Lossless compression of hyperspectral images using recursive least square lattice filter group. *Opt. Precis. Eng.* **2021**, *29*, 896. [CrossRef]
7. Altamimi, A.; Ben Youssef, B. A Systematic Review of Hardware-Accelerated Compression of Remotely Sensed Hyperspectral Images. *Sensors* **2022**, *22*, 263. [CrossRef]
8. Alvarez-Cortes, S.; Serra-Sagrista, J.; Bartrina-Rapesta, J.; Marcellin, M.W. Regression Wavelet Analysis for Near-Lossless Remote Sensing Data Compression. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 790–798. [CrossRef]
9. Roger, R.; Arnold, J. Reversible image compression bounded by noise. *IEEE Trans. Geosci. Remote Sens.* **1994**, *32*, 19–24. [CrossRef]
10. Blanes, I.; Magli, E.; Serra-Sagrista, J. A Tutorial on Image Compression for Optical Space Imaging Systems. *IEEE Geosci. Remote Sens. Mag.* **2014**, *2*, 8–26. [CrossRef]

11. Dua, Y.; Kumar, V.; Singh, R.S. Comprehensive review of hyperspectral image compression algorithms. *Opt. Eng.* **2020**, *59*, 090902. [CrossRef]
12. Valsesia, D.; Magli, E. Fast and Lightweight Rate Control for Onboard Predictive Coding of Hyperspectral Images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 394–398. [CrossRef]
13. Valsesia, D.; Magli, E. A Novel Rate Control Algorithm for Onboard Predictive Coding of Multispectral and Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 6341–6355. [CrossRef]
14. Bartrina-Rapesta, J.; Marcellin, M.W.; Serra-Sagrista, J.; Hernandez-Cabronero, M. A Novel Rate-Control for Predictive Image Coding With Constant Quality. *IEEE Access* **2019**, *7*, 103918–103930. [CrossRef]
15. Weinberger, M.; Seroussi, G.; Sapiro, G. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Trans. Image Process.* **2000**, *9*, 1309–1324. [CrossRef]
16. Wu, X.; Memon, N. Context-based, adaptive, lossless image coding. *IEEE Trans. Commun.* **1997**, *4*, 437–444. [CrossRef]
17. Wu, X.; Memon, N. Context-based lossless interband compression - Extending CALIC. *IEEE Trans. Image Process.* **2000**, *9*, 994–1001. [CrossRef]
18. Magli, E.; Olmo, G.; Quacchio, E. Optimized Onboard Lossless and Near-Lossless Compression of Hyperspectral Data Using CALIC. *IEEE Geosci. Remote Sens. Lett.* **2004**, *1*, 21–25. [CrossRef]
19. Klimesh, M. Low-Complexity Lossless Compression of Hyperspectral Imagery via Adaptive Filtering. No. IPN-PR-42-163 2005, Volume 42–163, pp. 1–10. Available online: https://ipnpr.jpl.nasa.gov/progress_report/42-163/163H.pdf (accessed on 12 July 2022).
20. Consultative Committee for Space Data Systems. Lossless Multispectral & Hyperspectral Image Compression. 2012. Available online: https://public.ccsds.org/Pubs/123x0b1ec1s.pdf (accessed on 12 July 2022).
21. Bartrina-Rapesta, J.; Blanes, I.; Auli-Llinas, F.; Serra-Sagrista, J.; Sanchez, V.; Marcellin, M.W. A Lightweight Contextual Arithmetic Coder for On-Board Remote Sensing Data Compression. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4825–4835. [CrossRef]
22. Ansari, R.; Memon, N.D.; Ceran, E. Near-lossless image compression techniques. *J. Electron. Imaging* **1998**, *7*, 486–494.
23. Tai, S.C.; Kuo, T.M.; Ho, C.H.; Liao, T.W. A near-lossless compression method based on CCSDS for satellite images. In Proceedings of the 2012 International Symposium on Computer, Consumer and Control, Taichung, Taiwan, 4–6 June 2012; pp. 706–709.
24. Consultative Committee for Space Data Systems. Recommendation for Space Data System Standards; 2005. Available online: https://public.ccsds.org/Pubs/122x0b1c3s.pdf (accessed on 12 July 2022).
25. Valsesia, D.; Magli, E. High-Throughput Onboard Hyperspectral Image Compression with Ground-Based CNN Reconstruction. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9544–9553. [CrossRef]
26. Wu, X.; Bao, P. Near-lossless image compression by combining wavelets and CALIC. In Proceedings of the Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers (Cat. No. 97CB36136), Pacific Grove, CA, USA, 2–5 November 1997; Volume 2, pp. 1427–1431.
27. Beerten, J.; Blanes, I.; Serra-Sagrista, J. A Fully Embedded Two-Stage Coder for Hyperspectral Near-Lossless Compression. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1775–1779. [CrossRef]
28. Dua, Y.; Singh, R.S.; Kumar, V. Compression of multi-temporal hyperspectral images based on RLS filter. *Vis. Comput.* **2022**, *38*, 65–75. [CrossRef]
29. CCSDS. 123.0-B-Info, TestData. 2019. Available online: http://cwe.ccsds.org/sls/docs/SLS-DC/123.0-B-Info/TestData (accessed on 1 June 2019).
30. Ibn Afjal, M.; Al Mamun, M.; Uddin, M.P. Band reordering heuristics for lossless satellite image compression with 3D-CALIC and CCSDS. *J. Vis. Commun. Image Represent.* **2019**, *59*, 514–526. [CrossRef]
31. Karaca, A.C.; Gullu, M.K. Superpixel based recursive least-squares method for lossless compression of hyperspectral images. *Multidimens. Syst. Signal Process.* **2019**, *30*, 903–919. [CrossRef]

remote sensing

MDPI

Article

Multi-Parameter Inversion of AIEM by Using Bi-Directional Deep Neural Network

Yu Wang [1], Zi He [1,*], Ying Yang [1], Dazhi Ding [1], Fan Ding [2] and Xun-Wang Dang [3]

[1] Department of Communication Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; wangyu409@njust.edu.cn (Y.W.); yangying@njust.edu.cn (Y.Y.); dzding@njust.edu.cn (D.D.)
[2] China Ship Development and Design Centre, Wuhan 430064, China; 10725@hbuas.edu.cn
[3] Science and Technology on Electromagnetic Scattering Laboratory, Beijing 100089, China; dangxunwang_207@casic.com.cn
* Correspondence: zihe@njust.edu.cn

Abstract: A novel multi-parameter inversion method is proposed for the Advanced Integral Equation Model (AIEM) by using bi-directional deep neural network. There is a very complex nonlinear relationship between the surface parameters (dielectric constant and roughness) and radar backscattering coefficient. The traditional inverse neural network, which is constructed by using the backscattering coefficients as the input and the surface parameters as the output, leads to bad convergence and wrong results. This is because many sets of surface parameters can get the same backscattering coefficient. Therefore, the proposed bi-directional deep neural network starts with building an AIEM-based forward deep neural network (AIEM-FDNN), whose inputs are the surface parameters and outputs are the backscattering coefficients. In this way, the weights and biases of the forward deep neural network can be optimized and predicted, which can be used for the backward deep neural network (AIEM-BDNN). Then, the multi-parameters are updated by minimizing the loss between the output backscattering coefficients with the measured ones. By inserting a sigmoid function between the input and the first hidden layer, the input multi-parameters can be efficiently approximated and continuously updated. As a result, both the forward and backward deep neural networks can be built with these weights and biases. By sharing the weights and biases of the forward network, the training of the inverse network is avoided. The bi-directional deep neural network can not only predict the backscattering coefficient but can also inverse the surface parameters. Numerical results are given to demonstrate that the RMSE of the backscattering coefficients calculated by the proposed bi-directional neural network can be reduced to 0.1%. The accuracy of the inversion parameters, including the real and imaginary parts of the dielectric constant, the root mean square height and the correlation length, can be improved to 97.56%, 91.14%, 99.04% and 98.45%, respectively. At the same time, the bi-directional neural network also has good accuracy for the inversion of the POLARSCAT measured data.

Keywords: bi-directional neural network; AIEM; surface parameters; backscattering coefficients

Citation: Wang, Y.; He, Z.; Yang, Y.; Ding, D.; Ding, F.; Dang, X.-W. Multi-Parameter Inversion of AIEM by Using Bi-Directional Deep Neural Network. *Remote Sens.* **2022**, *14*, 3302. https://doi.org/10.3390/rs14143302

Academic Editor: Yue Wu

Received: 31 May 2022
Accepted: 5 July 2022
Published: 8 July 2022

1. Introduction

The inversion of the surface parameters is the key problem in remote sensing science research [1–4]. Surface parameters can effectively reflect environmental conditions and understand the dynamic information for Earth monitoring. Therefore, there is a great significance in obtaining the surface parameters. Surface parameters inversion is to solve or calculate the target parameters that describe the actual situation of landforms according to the observation information and the forward physical model. How to combine the numerical and experimental results has always been a hot research topic. It has an important guiding significance for overland disturbances and environmental monitors. The inversion of the actual surface information is usually based on a random rough surface scattering

model. Over the past few decades, many researchers focused on surface scattering characteristics by using experimental and theoretical methods. The Kirchhoff Approximation (KA) was mostly applied to large-scale rough surfaces [5,6]. On the other hand, for small-scale rough surfaces, the Small Perturbation Model (SPM) was developed [7,8]. Subsequently, the Small Slope Approximation (SSA), which was proposed by Voronovich, combines the perturbation theory with the tangent approximation [9,10]. It should be noted that the KA is only suitable for a large curvature, while the SPM is only suitable for small roughness. Therefore, the Integral Equation Model (IEM) [11] was proposed by Fung to bridge the KA and SPM. The dependence of the surface height on the phase of the Green's function was ignored for the traditional IEM, which led to a big error. Then, a series of modified schemes were proposed to increase the accuracy, such as the Advanced Integral Equation Model (AIEM) and its derivatives [12,13]. Therefore, the AIEM can be used as an efficient tool to model the landform for its robustness and scalability.

The research methods in this area are generally divided into the empirical formula method, intelligent optimization algorithm and neural network method. In the past decades, the semi-empirical models were used as one of the most popular methods to predict the parameters [14–16]. This method is to summarize the laws of a large number of measured data and express them with simple functions. Inspired by evolutionary phenomena in nature, many intelligent optimization algorithms have emerged, such as the GA (Genetic Algorithm) and PSO (Particle Swarm Optimization). Such methods have been widely used for hydrogeological parameters and rough surface parameters inversion [17,18]. The core idea of the intelligent optimization algorithm is to use the algorithm to traverse the model space constructed by all the parameters to obtain the optimal solution of the objective function. However, it is difficult to obtain the global optimal solution using these methods, and only a small number of parameters can be inverted. At present, neural networks are being widely used in engineering fields such as machinery, materials and architecture, and their applications can be traced back to the late 1980s. Neural networks can perform complex data processing and are usually used to complete classification tasks and function approximation tasks. Therefore, a neural network is a promising tool for solving the inverse problems arising from its generalization ability. In [3], a back propagation Neural Network (BP) based on IEM was developed to inverse the surface parameters. In [19,20], neural networks with different structures were used for the prediction of metasurface geometric parameters or color parameters. Meanwhile, a Convolutional Neural Network (CNN) has been used in SAR target recognition and terrain classification [21–23]. In [24], a CNN and Generative Adversarial Network (GAN) were combined to extract simulation parameters from SAR images.

In this paper, a novel bi-directional DNN (deep neural network) is proposed to predict the multi-parameters of the AIEM. The proposed bi-directional DNN consists of two DNNs. Both DNNs share the same network structure and the same set of network weights. The bi-directional DNN can successfully complete the two tasks of predicting backscattering coefficients and inverting surface parameters. At first, a forward DNN needs to be established. This forward DNN takes the surface parameters as the input and the backscattering coefficients as the output. After training, this network can fit the AIEM model well. Then, a backward DNN is constructed by reusing the network structure of the forward DNN and the weights after training. Before backward network training, the input surface parameters need to be initialized as constants. Finally, the initialized surface parameters can be updated by calculating the loss of the output backscattering coefficients and the actual backscattering coefficients. The traditional inverse neural network, which is constructed by using the backscattering coefficients as the input and the surface parameters as the output, leads to a bad convergence and wrong result. However, the proposed bi-directional deep neural network is proposed to overcome these problems. Compared with the BP neural network, the proposed bi-directional network has a higher inversion accuracy. To verify the inversion accuracy of the bi-directional network, POLARSCAT [25–27] measured data on bare soil surfaces under three different roughness and humidity conditions was used.

The numerical results showed that the bi-directional network has high accuracy for the prediction of backscattering coefficients and the inversion of surface parameters.

2. Materials and Methods

2.1. Experimental Data

In this study, the training data of the bi-directional network was obtained based on the mapping relationship between the surface parameters and radar observations. In fact, training data satisfying such conditions cannot be obtained from the point datasets measured in the field. The AIEM model can simulate the backscattering characteristics under various surface conditions. Given the range of variations in the surface permittivity, the root mean square height (RMS) height and correlation length of interest bi-directional neural network training data can be generated by the AIEM model [12,28].

The general formula of the AIEM model is shown in Figure 1.

$$\sigma_{qp}(s) = \sigma_{qp}^{k} + \sigma_{qp}^{kc} + \sigma_{qp}^{c} \tag{1}$$

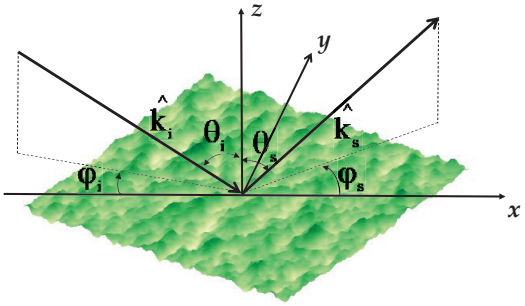


Figure 1. Schematic diagram of scattering from rough surfaces.

It can be seen that the scattering coefficient is composed of Kirchhoff terms σ_{qp}^{k}, cross terms σ_{qp}^{kc} and compensation terms σ_{qp}^{c}. The explicit form of AIEM can be given as

$$\begin{aligned}\sigma_{qp}(s) = \; & \frac{k^2}{2} e^{-\sigma^2(k_{sz}^2 + k_z^2)} \\ & \cdot \sum_{n=1}^{\infty} \frac{\sigma^{2n}}{n!} \left| I_{qp}^{n} \right|^2 S^{(n)}(k_{sx} - k_x, k_{sy} - k_y)\end{aligned} \tag{2}$$

where k is the incident wave number, σ^2 represents the variance of the surface height and $S^{(n)}\left(k_{sx} - k_x, k_{sy} - k_y\right)$ denotes the surface roughness spectrum of the surface in terms of the nth power of the surface correlation function by two-dimensional Fourier transform.

As shown in Figure 1, the incident and scattered wave vectors can be defined as

$$k_x = k \sin\theta_i \cos\varphi_i \; ; k_y = k \sin\theta_i \sin\varphi_i ; k_z = -k \cos\theta_i \tag{3}$$

$$k_{sx} = k \sin\theta_s \cos\varphi_s \; ; k_{sy} = k \sin\theta_s \sin\varphi_s ; k_{sz} = k \cos\theta_s \tag{4}$$

where θ_i and φ_i are the incident angle, and θ_s and φ_s are the scattering angle. The backscattering direction is at $\theta_i = \theta_s$, $\varphi_s = \varphi_i + 180^\circ$.

POLARSCAT is a polarizing scatterometer that operates on different bare surfaces, each with wet and dry conditions. The polarimetric measurements are conducted at the L-, C- and X-band frequencies at incident angles ranging from 10° to 70°. In this paper, the experimental data in the L- (the center frequency is 1.5 GHz) and X-bands (the center frequency is 4.75 GHz) are selected. As shown in Table 1, three soils of different roughness were measured in dry and wet conditions. Where σ is the RMS height, l is the correlation length and $k = 2\pi/\lambda$, ($\lambda = c/f, c = 3 \times 10^8$). The RMS height ranged from 0.40 cm to

1.12 cm, and the correlation length ranged from 8.4 cm to 9.9 cm. In [25–27], for the three surfaces (S1–S3), the measured autocorrelation function was found to be closer in shape to an exponential function.

Table 1. POLARSCAT measured parameters.

Surface Number	Freq. (GHz)	$k\sigma$	kl	$\sigma\ (cm)$	$l\ (cm)$	ε_r	ε_r''
S1-dry	1.5 GHz	0.13	2.6	0.40	8.4	7.99	2.02
	4.75 GHz	0.4	8.4			8.77	1.04
S1-wet	1.5 GHz	0.13	2.6			15.57	3.71
	4.75 GHz	0.4	8.4			15.42	2.15
S2-dry	1.5 GHz	0.1	3.1	0.32	9.9	5.85	1.46
	4.75 GHz	0.32	9.8			6.66	0.68
S2-wet	1.5 GHz	0.1	3.1			14.43	3.47
	4.75 GHz	0.32	9.8			14.47	1.99
S3-dry	1.5 GHz	0.35	2.6	1.12	8.4	7.70	1.95
	4.75 GHz	1.11	8.4			8.50	1.00
S3-wet	1.5 GHz	0.35	2.6			15.34	3.66
	4.75 GHz	1.11	8.4			15.23	2.12

2.2. Method

In this section, it will be introduced separately from the overall framework of the bi-directional network, the structure of the forward network and the structure of the reverse network. At the same time, the workflow of the bi-directional network will be introduced in detail.

2.2.1. Framework of the Bi-Directional Deep Neural Network

There are usually two smart methods for solving inverse problems, namely the optimization algorithm and neural network inverse modeling method. The core idea of the optimization algorithm is to traverse the model space constructed by all parameters to obtain the optimal solution of the objective function. However, this kind of method needs to manually set the range of each parameter, and it is easy to fall into the local optimal solution when dealing with complex problems. In [29], a genetic algorithm was used to invert the surface parameters. It is often necessary to perform multiple searches to select the optimal solution, and the accuracy is not high. Another method is to use the backscattering coefficients as the input and the surface parameters as the output and use the neural network to directly construct the inverse model. However, since there is no exact analytical formula from the backscattering coefficients to the surface parameters, at the same time, the non-uniqueness of the dataset itself will make the overall training of the dataset difficult for the inverse model, thus affecting the inversion accuracy.

In this paper, a novel DNN-based surface parameters inversion method is proposed. As shown in Figure 2, this framework consists of two DNNs, namely an AIEM-Based Forward Deep Neural Network (AIEM-FDNN) and AIEM-Based Backward Deep Neural Network (AIEM-BDNN). The same network structure and weights are shared by them. The AIEM-FDNN takes the surface parameters as the input and the backscattering coefficients as the output. After training, it can be used to quickly calculate the backscattering coefficients outside the dataset. The AIEM-BDNN can be formed by reusing the network structure and well-trained weights of AIEM-FDNN. The input nodes need to be set as the variables. First, the input surface parameters are randomly initialized as constants. Then, the loss between the output backscattering coefficients and the actual backscattering coefficients will be calculated by the AIEM-BDNN. Finally, based on the back propagation of the error, the initialized surface parameters are continuously updated by the optimizer until the error converges into a sufficiently small value. Meanwhile, the updated surface parameters are the inversion results of the AIEM-BDNN based on this set of backscattering coefficients.

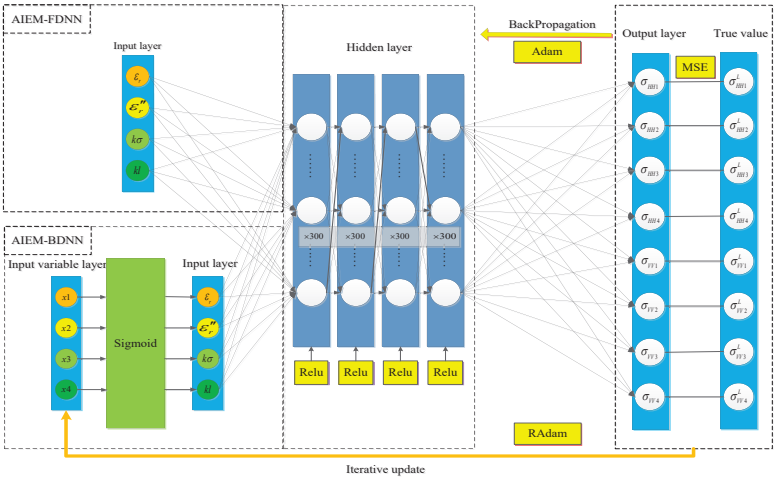


Figure 2. The framework for the proposed AIEM-based bi-directional deep neural network.

The flowchart of the overall working process of the bi-directional deep neural network is provided in Figure 3. The workflow of the AIEM-FDNN and AIEM-BDNN will be presented in detail in the following two parts.

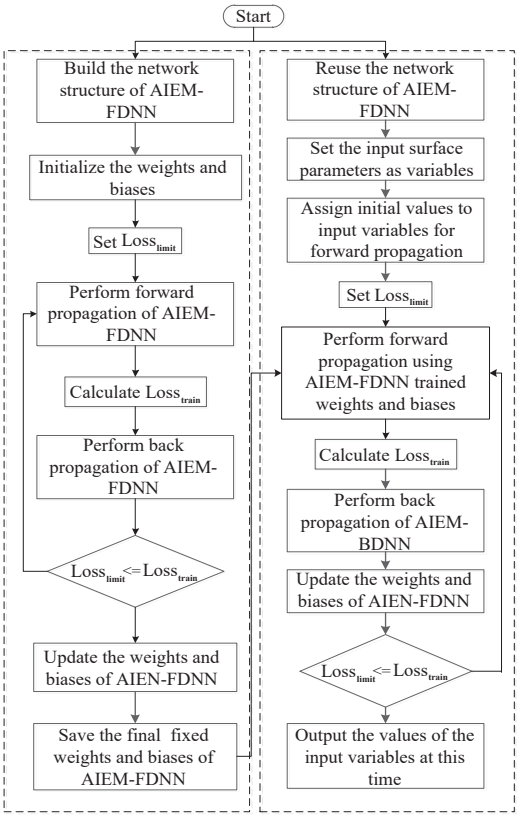


Figure 3. Flowchart of the working process of the bi-directional DNN.

2.2.2. AIEM-Based Forward Deep Neural Network

As shown in Figure 1, the AIEM-FDNN is a fully connected network that contains an input layer, multiple hidden layers and an output layer. Its input is the surface parameters, including the real part ε_r and imaginary part ε_r'' of the dielectric constant, the root mean square height $k\sigma$ and the correlation length kl, and the output is the backscattering coefficients σ_{HH}, σ_{VV}.

The AIEM-FDNN is designed to calculate the backscattering coefficients. The trained AIEM-FDNN has similar computational accuracy to the AIEM model, and it is less complex to calculate. Since the AIEM-BDNN used for surface parameters inversion uses the network structure of AIEM-FDNN and the weights after training, the accuracy of the AIEM-FDNN directly affects the performance of the entire bi-directional DNN. The training process of the AIEM-FDNN consists of two stages: forward propagation and back propagation. The forward propagation is to calculate the loss of the output backscattering coefficients and the actual backscattering coefficients according to the current network weights. The back propagation is to update the weights using gradient descent techniques based on the current loss.

The forward propagation calculation process of AIEM-FDNN can be given as

$$\mathbf{Z_{AF}^0} = [\varepsilon_r, \varepsilon_r'', k\sigma, kl]_A \tag{5}$$

$$\mathbf{Z_{AF}^i} = g_i\left(\mathbf{W_{AF}^i} \cdot \mathbf{Z_{AF}^{i\text{-}1}} + \mathbf{b_{AF}^i}\right)(i = 1, \cdots, N) \tag{6}$$

$$\mathbf{Z_{AF}^N} = g_N\left(\mathbf{W_{AF}^N} \cdot \mathbf{Z_{AF}^{N\text{-}1}} + \mathbf{b_{AF}^N}\right) \tag{7}$$

$$[\boldsymbol{\sigma_{HH}, \sigma_{VV}}] = \mathbf{Z_{AF}^N} \tag{8}$$

where $\mathbf{Z_{AF}^0}$ and $\mathbf{Z_{AF}^N}$ represent the input surface parameters and the output backscattering coefficients for HH and VV polarizations for different incident angles, respectively. $\mathbf{Z_{AF}^i}(i = 1, \cdots, N)$ represents the calculation result of the ith layer after the activation function. $\mathbf{W_{AF}^i}$ represents the weights matrix from the *(i-1)*th layer to the ith layer. $\mathbf{b_{AF}^i}$ represents the biases of the ith layer, and $g_i(\cdot)$ represents the nonlinear activation function of the ith layer. As shown in Figure 3, the calculated loss between the output backscattering coefficients and the actual backscattering coefficients will be calculated. The loss function of AIEM-FDNN is defined as the mean square error, which can be expressed as

$$Loss_{AF} = \frac{1}{n}\sum_{j=1}^{n}\left[\left(\sigma_{HH,j}^L - \sigma_{HH,j}\right)^2 + \left(\sigma_{VV,j}^L - \sigma_{VV,j}\right)^2\right] \tag{9}$$

where $\sigma_{HH,j}^L, \sigma_{VV,j}^L$ represents the actual backscattering coefficients of the *HH* and *VV* polarization for the jth incident angle. The back propagation of AIEM-FDNN is based on the chain derivation rule. $\frac{\partial Loss_{AF}}{\partial \mathbf{W_{AF}^i}}$ and $\frac{\partial Loss_{AF}}{\partial \mathbf{b_{AF}^i}}$ are calculated to update $\mathbf{W_{AF}^i}$ and $\mathbf{b_{AF}^i}$ until $Loss_{AF}$ converges to a minimum. The calculation process can be given as

$$\mathbf{E_{AF}^N} = -\left(\mathbf{y_{label}} - g_N\begin{pmatrix}\mathbf{W_{AF}^N} \cdot \mathbf{Z_{AF}^{N\text{-}1}} \\ +\mathbf{b_{AF}^N}\end{pmatrix}\right) \circ {g'}_N\begin{pmatrix}\mathbf{W_{AF}^N} \cdot \mathbf{Z_{AF}^{N\text{-}1}} \\ +\mathbf{b_{AF}^N}\end{pmatrix} \tag{10}$$

$$\mathbf{E_{AF}^i} = \left(\left(\mathbf{W_{AF}^{i+1}}\right)^T \cdot \mathbf{E_{AF}^{i+1}}\right) \circ {g'}_i\begin{pmatrix}\mathbf{W_{AF}^i} \cdot \mathbf{Z_{AF}^{i\text{-}1}} \\ +\mathbf{b_{AF}^i}\end{pmatrix} \tag{11}$$

$$\frac{\partial Loss_{AF}}{\partial \mathbf{W_{AF}^i}} = \mathbf{E_{AF}^i} \cdot \left(\mathbf{Z_{AF}^{i-1}}\right)^T \tag{12}$$

$$\frac{\partial Loss_{AF}}{\partial \mathbf{b_{AF}^i}} = \mathbf{E_{AF}^i} \tag{13}$$

where $\mathbf{E_{AF}^{N}}$ represents the error vector in the output layer of AIEM-FDNN, ${g'}_i(\cdot)$ is the derivative of the activation function. $\mathbf{E_{AF}^{i}}$ is the error vector in the ith layer, and $\circ$ is the Hadamard product. Finally, the formulas for updating the weights and biases can be given as

$$\mathbf{W_{AF}^{i}} = \mathbf{W_{AF}^{i}} - \eta_{AF}\frac{\partial Loss_{AF}}{\partial \mathbf{W_{AF}^{i}}} \tag{14}$$

$$\mathbf{b_{AF}^{i}} = \mathbf{b_{AF}^{i}} - \eta_{AF}\frac{\partial Loss_{AF}}{\partial \mathbf{b_{AF}^{i}}} \tag{15}$$

where η_{AF} represents the learning rate of the AIEM-FDNN.

2.2.3. AIEM-Based Backward Deep Neural Network

The AIEM-BDNN is constructed by directly reusing the network structure of the AIEM-FDNN and loading the training weights and biases to invert the surface parameters. Simply put, it is only necessary to set the input node of the trained AIEM-FDNN as variables. The training process of the AIEM-BDNN also includes forward propagation and back propagation, but it is different from the training object of AIEM-FDNN. The training objects of AIEM-FDNN are the weights and biases of the network, while the training objects of the AIEM-BDNN are the input surface parameters of the network. The AIEM-BDNN is trained by giving a set of backscattering coefficients to be inverted. By initializing the input surface parameters as constants, the forward propagation of the AIEM-BDNN is performed to calculate the backscattering coefficients. Back propagation is performed according to the loss between the output backscattering coefficients and the true backscattering coefficients. Finally, the initialized surface parameters are continuously updated until the loss converges to a small enough value. The last surface parameters updated are the inversion values.

The forward propagation calculation process of the AIEM-FDNN can be given as

$$\mathbf{Z_{AB}^{0}} = [\varepsilon_r, \varepsilon_r'', k\sigma, kl]_B \tag{16}$$

$$\mathbf{Z_{AB}^{i}} = g_i\left(\mathbf{W_{AF}^{i}} \cdot \mathbf{Z_{AB}^{i\text{-}1}} + \mathbf{b_{AF}^{i}}\right)(i = 1, \cdots, N) \tag{17}$$

$$\mathbf{Z_{AB}^{N}} = g_N\left(\mathbf{W_{AF}^{N}} \cdot \mathbf{Z_{AB}^{N-1}} + \mathbf{b_{AF}^{N}}\right) \tag{18}$$

$$[\boldsymbol{\sigma_{HH}}, \boldsymbol{\sigma_{VV}}] = \mathbf{Z_{AB}^{N}} \tag{19}$$

where $[\varepsilon_r, \varepsilon_r'', k\sigma, kl]_B$ are randomly initialized surface parameters. $\mathbf{W_{AF}^{i}}$ represents the weights matrix from the *(i-1)*th layer to the ith layer of AIEM-FDNN. $\mathbf{b_{AF}^{i}}$ represents the biases of the ith layer of the AIEM-FDNN. Since the AIEM-FDNN has been trained, $\mathbf{W_{AF}^{i}}$ and $\mathbf{b_{AF}^{i}}$ have been fixed. They will not be updated in both the forward and backward propagation of the AIEM-BDNN. $g_i(\cdot)$ represents the nonlinear activation function of the ith layer of the AIEM-FDNN. $\mathbf{Z_{AB}^{i}}(i = 1, \cdots, N)$ represents the calculation results of the ith layer of the AIEM-BDNN after the activation function. The loss function of the AIEM-BDNN is also defined as the mean squared error, which can be expressed as

$$Loss_{AB} = \frac{1}{n}\sum_{j=1}^{n}\left[\left(\sigma_{HH,j}^{L} - \sigma_{HH,j}\right)^2 + \left(\sigma_{VV,j}^{L} - \sigma_{VV,j}\right)^2\right] \tag{20}$$

The back propagation of the AIEM-BDNN is also based on the chain derivation rule. $\frac{\partial Loss_{AB}}{\partial \mathbf{Z_{AB}^{0}}}$ is calculated to update $\mathbf{Z_{AB}^{0}}$ until $Loss_{AB}$ converges to a minimum. The calculation process can be given as

$$\mathbf{E_{AB}^{N}} = -\left(\mathbf{y_{label}} - g_N\left(\begin{array}{c}\mathbf{W_{AF}^{N}} \cdot \mathbf{Z_{AB}^{N-1}} \\ +\mathbf{b_{AF}^{N}}\end{array}\right)\right) \circ {g'}_N\left(\begin{array}{c}\mathbf{W_{AF}^{N}} \cdot \mathbf{Z_{AB}^{N-1}} \\ +\mathbf{b_{AF}^{N}}\end{array}\right) \tag{21}$$

$$\mathbf{E}_{\mathbf{AB}}^{\mathbf{i}} = \left(\left(\mathbf{W}_{\mathbf{AF}}^{\mathbf{i+1}}\right)^{T} \cdot \mathbf{E}_{\mathbf{AB}}^{\mathbf{i+1}}\right) \circ g'_{i}\left(\begin{array}{c}\mathbf{W}_{\mathbf{AF}}^{\mathbf{i}} \cdot \mathbf{Z}_{\mathbf{AB}}^{\mathbf{i-1}} \\ +\mathbf{b}_{\mathbf{AF}}^{\mathbf{i}}\end{array}\right) \tag{22}$$

$$\frac{\partial Loss_{AB}}{\partial \mathbf{Z}_{\mathbf{AB}}^{0}} = \left(\mathbf{W}_{\mathbf{AF}}^{1}\right)^{T} \cdot \mathbf{E}_{\mathbf{AB}}^{1} \tag{23}$$

where $\mathbf{E}_{\mathbf{AB}}^{\mathbf{N}}$ represents the error vector in the output layer of the AIEM-BDNN, and $g'_{i}(\cdot)$ is the derivative of activation function of AIEM-FDNN. $\mathbf{E}_{\mathbf{AB}}^{\mathbf{i}}$ is the error vector in the ith layer, and $\circ$ is the Hadamard product. Finally, the formulas for updating $\mathbf{Z}_{\mathbf{AB}}^{0}$ can be given as

$$\mathbf{Z}_{\mathbf{AB}}^{0} = \mathbf{Z}_{\mathbf{AB}}^{0} - \eta_{AB}\frac{\partial Loss_{AB}}{\partial \mathbf{Z}_{\mathbf{AB}}^{0}} \tag{24}$$

in which η_{AB} represents the learning rate of the AIEM-BDNN.

From the formula derivation of AIEM-FDNN and AIEM-BDNN, it can be seen that the training purpose of the AIEM-FDNN is to update the weights and biases of the network. Instead, AIEM-BDNN uses the weights and biases that AIEM-FDNN has already trained and fixed. Therefore, its training purpose is only to update the input parameters. It can be seen that the AIEM-FDNN and AIEM-BDNN are closely related. The quality of the AIEM-FDNN training will directly affect the inversion accuracy of the AIEM-BDNN. Therefore, using the bi-directional network to invert the surface parameters, we first need to ensure that the accuracy of the backscattering coefficients calculated by the AIEM-FDNN is high enough. The pseudocode of the bi-directional deep neural network was added as Appendix A to the article.

3. Results

3.1. Performance of the AIEM-Based Forward Deep Neural Network

The selection of the datasets is crucial for the training of neural networks. Since the AIEM model can simulate the backscattering characteristics under various surface parameters, the training set required for the AIEM-FDNN can be generated as long as the variation range of the surface parameters is given. As shown in Table 2, the range of each surface parameter for generating the dataset is given. The range of the radar incident angle is set from 20° to 50°. Four surface parameters, namely the real and imaginary parts of the dielectric constant, the normalized root mean square height and the normalized correlation length, are used as the input of the AIEM-FDNN, while the backscattering coefficients for HH and VV polarization are the output. The sampling interval of the real part and imaginary part of the dielectric constant is 1.2 and 1, respectively. The sampling interval of the normalized root mean square height is 0.1. The normalized relative length is 0.7. A number of (21,009) sets of surface parameter combinations were generated by a cyclic combination within the range of surface parameters, and the corresponding backscattering coefficients were calculated by using the AIEM model. Many (3000) groups were selected as the validation set, and 1300 groups were selected as the test set.

Table 2. Surface parameters and radar parameters.

Parameter	Value
Real part of the dielectric constant(ε_r)	2–26
Imaginary part of the dielectric constant (ε_r'')	0.1–10.1
Normalized root mean square height height ($k\sigma$)	0.1–1
Normalized relative lenght (kl)	1–10.8
Range of incident angle (θ_i)	20°–50°
Polarization mode	HH, VV
$k\sigma/kl$	0.01–0.5
$\varepsilon_r''/\varepsilon_r$	0–0.5
Surface roughness spectrum (S)	Exponential

Next, the AIEM-FDNN is built for forward prediction. After continuous testing and adjustment of the hyperparameters, the hyperparameter settings shown in Table 3 are finally determined. There are four hidden layers added in the AIEM-FDNN, and each layer has 300 neurons. The activation function of each hidden layer adopts the ReLU function. Then, using the mean squared error (MSE) as the loss function, the error between the output value and the true value for each epoch is calculated. At the same time, the popular optimizer Adam is used to realize the back propagation. Finally, the continuous updating of the weights and the biases can be realized. A decaying learning rate is used, so that the training loss can converge more smoothly. Setting the batch size to 20, the network converges when the epoch is equal to 1300.

Table 3. Training the hyperparameters of the AIEM-FDNN.

Parameter	Value
Weight initialization method	Uniform distribution initialization
Activation function	ReLU
Loss function	MSE
Optimizer	Adam
Learning rate	0.001
Learning decay rate	0.9
Hidden layers	4
Hidden neurons	300
Epoch	1300
Batch size	20

The test set was used to test the ability of the AIEM-FDNN to predict backscattering coefficients. As shown in Table 4, the RMSE between the output backscattering coefficients for HH and VV polarizations for different incident angles and the actual backscattering coefficients for HH and VV polarizations for different incident angles can be reduced to be less than 0.1%. It can be seen that the training of the AIEM-FDNN is successful, and the accuracy is high. The trained AIEM-FDNN has almost the same computational accuracy as the AIEM model. The 21,009 sets of data generated by the AIEM model need 75.6 s, with 7.34 s for the proposed AIEN-FDNN. Therefore, the AIEM-FDNN has a faster computation speed when faced with a large amount of data generation tasks.

Table 4. The RMSE between the output backscattering coefficients and the actual backscattering coefficients for the proposed AIEM-FDNN with $\varphi = 0^\circ - 180^\circ$.

Polarization	Incident Angle (θ)	RMSE
VV	20°	0.1055%
VV	30°	0.0585%
VV	40°	0.0557%
VV	50°	0.0708%
HH	20°	0.0905%
HH	30°	0.0589%
HH	40°	0.0661%
HH	50°	0.0655%

At the same time, the degree of agreement between the backscattering coefficients calculated by AIEM-FDNN and the measured data has a great influence on the accuracy of the bi-directional network inversion of actual surface parameters. POLARSCAT measured data are used to test the AIEM-FDNN. The comparison of backscattering coefficients of the AIEM (AIEM-VV and AIEM-HH), POLARSCAT measured data (data_VV and data_HH) and AIEM-FDNN (AIEM-FDNN_VV and AIEM-FDNN_HH) for exponential correlated surface are shown in Figure 4. It can be seen that the three have good consistency. This lays a good foundation for the AIEM-BDNN to invert POLARSCAT measured parameters.

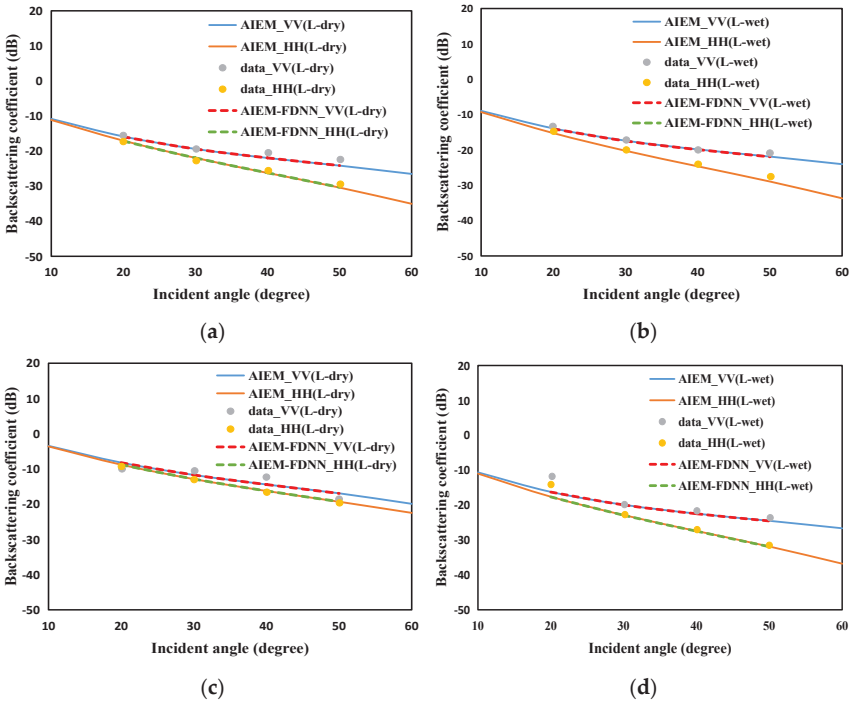


Figure 4. Comparison of backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for exponential correlated surface with (**a**) $\varepsilon_r = 7.99$, $\varepsilon_r'' = 2.02$, $k\sigma = 0.13$ and $kl = 2.6$ at 1.5 GHz; (**b**) $\varepsilon_r = 15.57$, $\varepsilon_r'' = 3.71$, $k\sigma = 0.13$ and $kl = 2.6$ at 1.5 GHz; (**c**) $\varepsilon_r = 7.7$, $\varepsilon_r'' = 1.95$, $k\sigma = 0.35$, $kl = 2.6$ at 1.5 GHz and (**d**) $\varepsilon_r = 14.43$, $\varepsilon_r'' = 3..47$, $k\sigma = 0.1$ and $kl = 3.1$ at 1.5 GHz.

3.2. Performance of the AIEM-Based Backward Deep Neural Network

AIEM-BDNN is designed to complete the surface parameters inversion task. It can be established by reusing the network structure of the AIEM-FDNN and the well-trained weights and biases. It is worth noting that the weights and biases of the AIEM-FDNN have been fixed and will not change after being reused by the AIEM-BDNN. Simply put, only the input surface parameters of the AIEM-BDNN are updated during training. The hyperparameters used by the AIEM-FDNN are not suitable for the AIEM-BDNN. After continuous tuning, the RAdam optimizer was chosen instead of the Adam optimizer. Xavier Initialization is chosen as the initialization method of the input surface parameters.

Two outstanding problems were found in the experiments, one of which is that the surface parameters are not updated in the desired direction. As a result, although the training loss can converge normally, the surface parameters obtained by the final inversion often deviate from the conventional parameter space. The update of the surface parameters is not automatically limited to the respective data ranges shown in Table 2, and even negative values may appear. The reason for this is that the AIEM-BDNN can accept arbitrary update parameters due to the training mechanism of the DNN, and even the wrong parameter combination can calculate the same result as the real value. In order to limit the update range of the input surface parameters, before AIEM-FDNN training, the input surface parameters are normalized by the method of Min–Max_scale, and the parameters can be limited to 0–1. Next, a sigmoid layer is inserted between the input layer and the first hidden layer of AIEM-BDNN. As a commonly used nonlinear function, the sigmoid function can limit any input value between 0 and 1. In this way, you do not need the need to care whether the updated surface parameters are out of a reasonable

range, because no matter how unreasonable the value of the updated surface parameter is, the sigmoid function will adjust it to the normal range. It should be noted here that, although the update object of the network is still the input surface parameters, the real input parameters of the AIEM-BDNN have become the values adjusted by the sigmoid function. At the same time, the value adjusted by the sigmoid function will also be used as the surface parameters inversed by AIEM-BDNN.

Another problem in the experiment is that there is a "premature" phenomenon when the input surface parameters are updating. This phenomenon is reflected in the fact that the training error cannot converge in the early stage of training. The reason is that, in the early stage of network training, the gradient decreases sharply, resulting in the slow update of neurons and ineffective learning. To alleviate such problem, the RAdam optimizer is used instead of the Adam optimizer, and the Xavier initialization method is used. The RAdam optimizer introduces a warm-up mechanism based on the commonly used Adam optimizer. Simply put, it is to use a small learning rate in the early stage of network training, so that the early training can be carried out smoothly and avoid excessive variance. The Xavier Initialization method will control the variance of the initial value within an appropriate range, usually making the variance of the initial value 1. It is also possible to choose to use the solution in [30,31]. By scanning all the variable hyperparameters in the AIEM-BDNN and recording the loss value, the one with the smallest loss is selected as the optimal inversion result. After continuous testing and adjustment of the hyperparameters, the hyperparameter settings shown in Table 5 are finally determined.

Table 5. Training hyper-parameters of AIEM-BDNN.

Parameter	Value
Input value initialization method	Xavier Initialization
Activation function	ReLU
Loss function	MSE
Optimizer	RAdam
Learning rate	0.001
Learning decay rate	0.9
Hidden layers	4
Hidden neurons	300
Epoch	10,000

Many (1300) sets of test sets are used to examine the inversion accuracy of the AIEM-BDNN. As shown in Figure 5, the comparison of the true surface parameters and the AIEM-BDNN predicted surface parameters is given. The numerical results show that the predicted surface parameters and the true surface parameters are concentrated near the contour, which shows that the accuracy of the predicted parameters is high. The correlation coefficient between the two is calculated, respectively, 97.56% (ε_r), 91.14% (ε_r''), 99.04% ($k\sigma$) and 98.45% (kl), as shown in Table 6.

Table 6. Inversion accuracy of the bi-directional neural networks.

Parameter	RMSE	Similarity(1-RMSE)
ε_r	0.0244	97.56%
ε_r''	0.0886	91.14%
$k\sigma$	0.0096	99.04%
kl	0.0155	98.45%

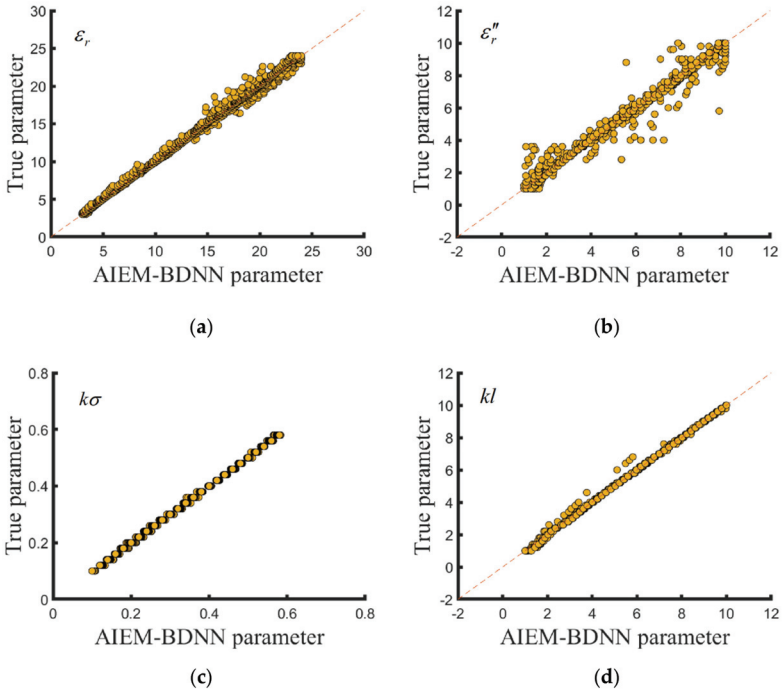


Figure 5. Comparison of true parameters and the AIEM−BDNN predicted parameters for: (**a**) the real part of the dielectric constant, (**b**) the imaginary part of the dielectric constant, (**c**) the normalized root mean square height and (**d**) the normalized correlation length.

As shown in Table 7, twelve sets of inversion results between POLARSCAT measured data and inverted by the AIEM-BDNN are compared. Three exponential distribution surfaces of POLARSCAT measured data are selected. As we can see, the comparison of the inversion results with the measured surface parameters can achieve satisfactory accuracy.

Table 7. Comparison of the surface parameters between POLARSCAT measured data and inverted by the AIEM-BDNN.

Surface Number	POLARSCAT (Measured)				AIEM-BDNN (Inverted)			
	ε_r	ε_r''	$k\sigma$	kl	ε_r	ε_r''	$k\sigma$	kl
S1-dry	7.99	2.02	0.13	2.6	9.07	1.23	0.13	2.81
	8.77	1.04	0.40	8.4	9.33	1.19	0.40	8.49
S1-wet	15.57	3.71	0.13	2.6	15.19	4.09	0.13	2.79
	15.42	2.15	0.40	8.4	16.00	0.36	0.40	8.44
S2-dry	5.85	1.46	0.10	3.1	3.02	2.96	0.16	1.29
	6.66	0.68	0.32	9.8	3.23	0.95	0.36	1.00
S2-wet	14.43	3.47	0.10	3.1	10.58	5.43	0.10	3.09
	14.47	1.99	0.32	9.8	14.91	1.63	0.32	9.88
S3-dry	7.7	1.95	0.35	2.6	7.41	2.53	0.31	1.89
	8.5	1.00	1.11	8.4	9.34	0.42	0.99	6.66
S3-wet	15.34	3.66	0.35	2.6	20.79	4.49	0.32	1.04
	15.23	2.12	1.11	8.4	15.00	4.58	0.99	8.86
		ε_r		ε_r''	$k\sigma$		kl	
RMSE		2.36		1.21	0.055		2.69	
nRMSE		0.1328		0.2386	0.0617		0.3029	

As shown in Figures 6–11 the inverted surface parameters are brought into the AIEM-FDNN. The obtained backscattering coefficients are compared with the measured values. It can be seen that the two have a good consistency.

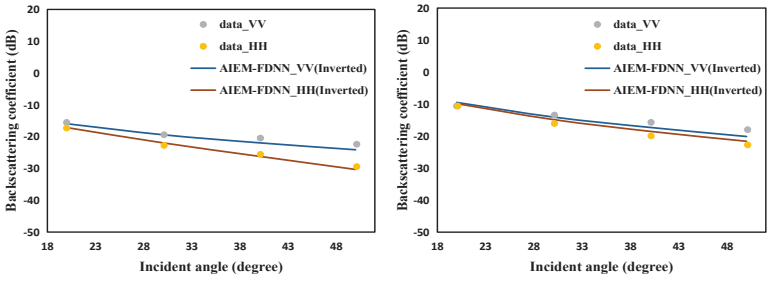


Figure 6. Comparison of the backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for exponential correlated surfaces with (**a**) measured: $\varepsilon_r = 7.99$, $\varepsilon_r'' = 2.02$, $k\sigma = 0.13$ and $kl = 2.6$ at 1.5 GHz; inverted: $\varepsilon_r = 9.07$, $\varepsilon_r'' = 1.23$, $k\sigma = 0.13$ and $kl = 2.81$; (**b**) measured: $\varepsilon_r = 8.77$, $\varepsilon_r'' = 1.04$, $k\sigma = 0.4$ and $kl = 8.4$ at 4.75 GHz; inverted: $\varepsilon_r = 9.33$, $\varepsilon_r'' = 1.19$, $k\sigma = 0.40$ and $kl = 8.49$.

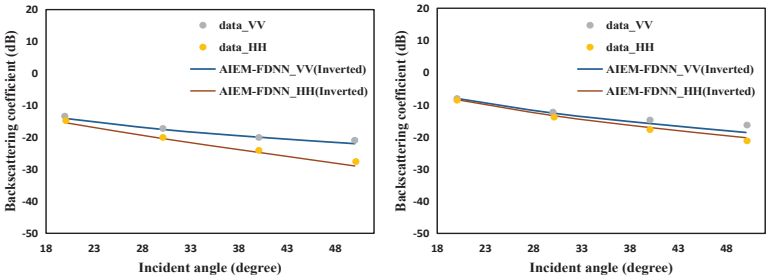


Figure 7. Comparison of the backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for the exponential correlated surface with (**a**) measured: $\varepsilon_r = 15.57$, $\varepsilon_r'' = 3.71$, $k\sigma = 0.13$, and $kl = 2.6$ at 1.5 GHz; inverted:$\varepsilon_r = 15.19$, $\varepsilon_r'' = 4.09$, $k\sigma = 0.13$ and $kl = 2.79$; (**b**) measured: $\varepsilon_r = 15.42$, $\varepsilon_r'' = 2.15$, $k\sigma = 0.40$ and $kl = 8.4$ at 4.75 GHz; inverted: $\varepsilon_r = 16.00$, $\varepsilon_r'' = 0.36$, $k\sigma = 0.40$ and $kl = 8.44$.

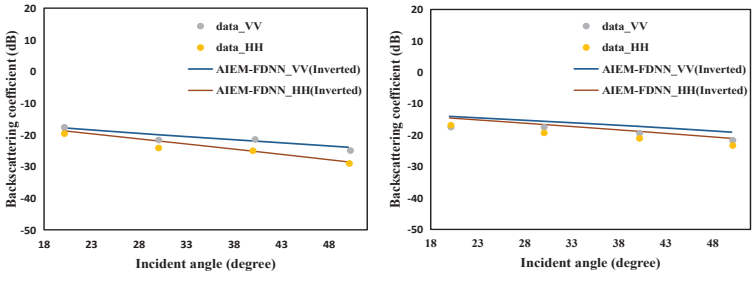


Figure 8. Comparison of backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for the exponential correlated surface with (**a**) measured: $\varepsilon_r = 5.85$, $\varepsilon_r'' = 1.46$, $k\sigma = 0.10$ and $kl = 3.1$ at 1.5 GHz; inverted: $\varepsilon_r = 3.02$, $\varepsilon_r'' = 2.96$, $k\sigma = 0.16$ and $kl = 1.29$; (**b**) measured: $\varepsilon_r = 6.66$, $\varepsilon_r'' = 0.68$, $k\sigma = 0.32$ and $kl = 9.8$ at 4.75 GHz; inverted: $\varepsilon_r = 3.23$, $\varepsilon_r'' = 0.95$, $k\sigma = 0.36$ and $kl = 1.00$.

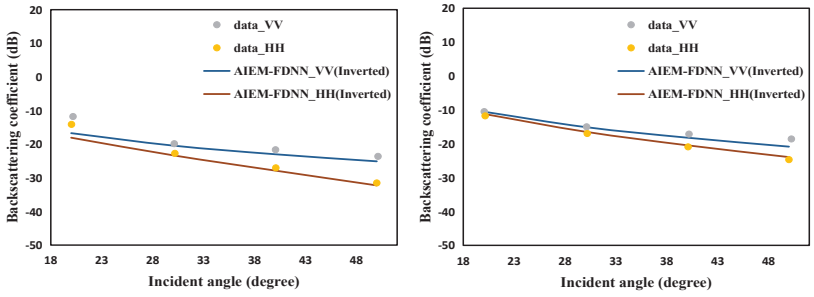


Figure 9. Comparison of the backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for the exponential correlated surface with (**a**) measured: $\varepsilon_r = 14.43$, $\varepsilon_r'' = 3.47$, $k\sigma = 0.10$ and $kl = 3.1$ at 1.5 GHz; inverted: $\varepsilon_r = 10.58$, $\varepsilon_r'' = 5.43$, $k\sigma = 0.10$ and $kl = 3.09$; (**b**) measured: $\varepsilon_r = 14.47$, $\varepsilon_r'' = 1.99$, $k\sigma = 0.32$ and $kl = 9.8$ at 4.75 GHz; inverted: $\varepsilon_r = 14.91$, $\varepsilon_r'' = 1.63$, $k\sigma = 0.32$ and $kl = 9.88$.

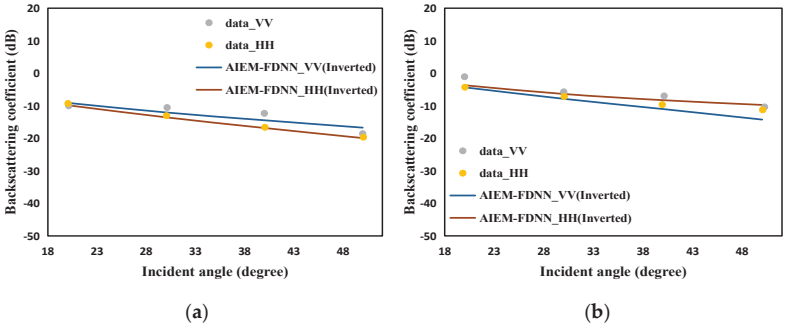


Figure 10. Comparison of the backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for the exponential correlated surface with (**a**) measured: $\varepsilon_r = 7.77$, $\varepsilon_r'' = 1.95$, $k\sigma = 0.35$ and $kl = 2.6$ at 1.5 GHz; inverted: $\varepsilon_r = 7.41$, $\varepsilon_r'' = 2.53$, $k\sigma = 0.31$ and $kl = 1.89$; (**b**) measured: $\varepsilon_r = 8.5$, $\varepsilon_r'' = 1.00$, $k\sigma = 1.11$ and $kl = 8.4$ at 4.75 GHz; inverted: $\varepsilon_r = 9.34$ $\varepsilon_r'' = 0.42$, $k\sigma = 0.99$ and $kl = 6.66$.

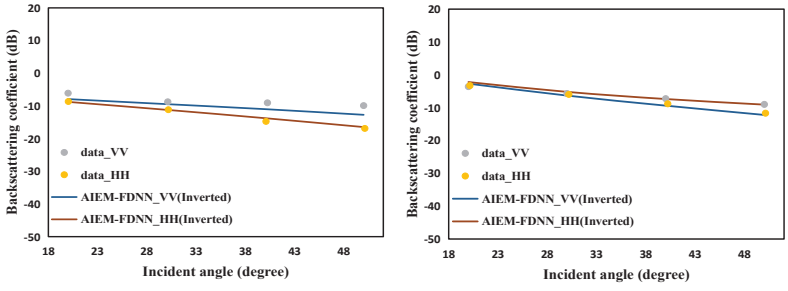


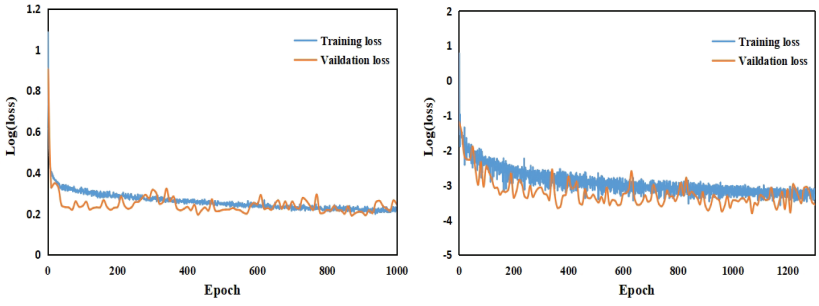
Figure 11. Comparison of the backscattering coefficients of the AIEM, POLARSCAT measured data and AIEM−FDNN for the exponential correlated surface with (**a**) measured: $\varepsilon_r = 15.34$, $\varepsilon_r'' = 3.66$, $k\sigma = 0.35$ and $kl = 2.6$ at 1.5 GHz; inverted: $\varepsilon_r = 20.79$, $\varepsilon_r'' = 4.49$, $k\sigma = 0.32$ and $kl = 1.04$; (**b**) measured: $\varepsilon_r = 15.23$, $\varepsilon_r'' = 2.12$, $k\sigma = 1.11$ and $kl = 8.4$ at 4.75 GHz; inverted: $\varepsilon_r = 15.00$, $\varepsilon_r'' = 4.58$, $k\sigma = 0.99$ and $kl = 8.86$.

4. Discussion

In this paper, the bi-directional network performs well in the task of surface parameter inversion. The bi-directional network has a high inversion accuracy for the AIEM model dataset. Similarly, for the inversion of the POLARSCAT measured data by the bi-directional network, the inversion value has a good correlation with the real value.

The bi-directional network is proposed to solve the problem of non-uniqueness, which leads to the poor effect of direct training of the inverse network. The nonunique data in the dataset itself will cause the training error of the directly constructed inverse network (with the backscattering coefficients as the input and the surface parameters as the output) to be unable to decrease and converge well. To solve this problem, bi-directional networks are proposed. The forward network AIEM-FDNN (with the surface parameters as the input and the backscattering coefficients as the output) is first trained, and the inverse network is constructed by reusing the weights trained by the AIEM-FDNN. In this way, the problem of directly constructing the inverse network can be avoided, and the bi-directional network achieves better inversion accuracy.

A BP (back propagation) neural network with backscattering coefficients as the input and surface parameters as the output is directly constructed. The 21,009 datasets generated by the AIEM model are used for training, and the training loss curve is shown in Figure 12a. Note that the training stops when the validation loss does not drop for 40 consecutive epochs. It can be seen that the training and validation losses for the BP neural network are 1.6257 and 1.5519, respectively, and the loss value barely dropped. This shows that the directly built inverse network performs poorly for the task of inverting surface parameters from input backscattering coefficients. The biggest reason that the inverse network cannot be trained well is the most common non-uniqueness problem in the inverse task of the neural network. Since the combination of different surface parameters can obtain the same or similar backscattering coefficients, this leads to a one-to-many situation during inverse network training. Once there are too many nonunique data in the dataset, the training loss of the network cannot be reduced well. On the contrary, the training of the forward network with the surface parameters as the input and the backscattering coefficient as the output does not have the influence of nonunique data on it. Therefore, it is hoped to start from the forward network and design a new method of surface parameter inversion. A bi-directional network was designed to overcome the above problems.



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

Figure 12. (**a**) Learning curve of the BP neural network. (**b**) Learning curve of the AIEM−FDNN.

As shown in Figure 12b, the loss curve of training and validation converges to a small value and keeps fluctuating after the AIEM-FDNN trained for 1300 epochs. Finally, the training loss value and validation loss value of the network are 6.45×10^{-4} and 1.17×10^{-4}, respectively. This loss value of the proposed bi-directional DNN is smaller than the traditional inverse network by several magnitudes. The weights trained by the AIEM-FDNN can be directly reused by the AIEM-BDNN, which can show a better loss convergence. As shown in Table 8, the bi-directional network achieves a better inversion accuracy.

Table 8. Inversion accuracy of BP neural networks and Bi-directional DNN.

	Bi-Directional DNN		BP	
Parameter	**RMSE**	**Similarity (1-RMSE)**	**RMSE**	**Similarity (1-RMSE)**
ε_r	0.0244	97.56%	0.0528	94.72%
ε_r''	0.0886	91.14%	0.4948	50.52%
$k\sigma$	0.0096	99.04%	0.0457	95.43%
kl	0.0155	98.45%	0.0374	96.26%

5. Conclusions

In this paper, a novel bi-directional neural network was proposed to invert the surface parameters. The establishment of the bi-directional network is divided into two steps. The AIEM-FDNN established first takes the surface parameters as the input and the backscattering coefficients as the output. The trained AIEM-FDNN can predict the backscattering coefficients outside the training dataset, and the predictions are also very accurate for the measured data. The AIEM-BDNN is built by reusing weights and biases trained by the AIEM-FDNN. At the same time, it is necessary to give the input surface parameter initialization constants, and a sigmoid layer between the input layer and the first hidden layer is inserted. After the error between the output backscattering coefficients and the true backscattering coefficients is continuously reduced, the input surface parameters can be continuously updated. The numerical results show that the bi-directional network not only has a good inversion effect for the data in the dataset but also has a high inversion accuracy for the measured data outside the dataset.

The bi-directional network is divided into a forward network (AIEM-FDNN) and an inverse network (AIEM-BDNN). The AIEM-BDNN is constructed by reusing the weights and biases of the AIEM-FDNN and does not require secondary training. Therefore, the training accuracy of the AIEM-FDNN will directly determine the inversion accuracy of the AIEM-BDNN. If the training effect of the forward network on some datasets is not good, then the bi-directional network will not be able to achieve a good inversion result.

One limitation we had to deal with in this paper is that the datasets used were only for backscattering coefficients under HH and VV polarizations. As a future work direction, we plan to incorporate the backscattering coefficients under HV and VH polarizations. The more abundant features of the four polarizations were used to further improve the accuracy of the surface parameters inversion. In addition, we considered adding part of the measured data to the dataset generated by the AIEM for training. We hope to reduce some of the differences between the simulated and measured data.

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

Funding: This work was supported in part by Natural Science 62071231, 61890541 and 61931021; Jiangsu Province Natural Science Foundation under Grant BK20211571 and the Fundamental Research Funds for the Central Universities of No. 30921011207, Laboratory of Pinghu (Beijing Institute of infinite electric Measurement), Science and Technology on Electromagnetic Scattering Laboratory.

Data Availability Statement: No applicable.

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

Appendix A

Algorithm A1: Bi-directional Deep Neural Network

Input: The input surface parameters $\mathbf{Z}^0_{\mathbf{AF}}$ and $\mathbf{Z}^0_{\mathbf{A}B}$, the true backscattering coefficients $[\sigma^L_{HH}, \sigma^L_{VV}]$, the maximum epoch I, the weights matrix $\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}}$ and $\mathbf{W}^{\mathbf{i}}_{\mathbf{A}B}$, the bias vector $\mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}$ and $\mathbf{b}^{\mathbf{i}}_{\mathbf{A}B}$, the nonlinear activation function $g_i(\cdot)$, the loss function MSE, the learning rate η_{AF}, η_{AB}

initialize $\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}}$ and $\mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}$
for $j = 1; j \leq I$ **do**
  $\mathbf{Z}^{\mathbf{i}}_{\mathbf{AF}} = g_i\left(\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}} \cdot \mathbf{Z}^{\mathbf{i\text{-}1}}_{\mathbf{AF}} + \mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}\right)(i = 1, \cdots, N)$
  $\mathbf{Z}^{N}_{\mathbf{AF}} = g_N\left(\mathbf{W}^{N}_{\mathbf{AF}} \cdot \mathbf{Z}^{N-1}_{\mathbf{AF}} + \mathbf{b}^{N}_{\mathbf{AF}}\right)$
  $Loss_{AF} = MSE(\mathbf{Z}^{N}_{\mathbf{AF}}, [\sigma^L_{HH}, \sigma^L_{VV}])$
  $\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}} = \mathbf{W}^{\mathbf{i}}_{\mathbf{AF}} - \eta_{AF}\frac{\partial Loss_{AF}}{\partial \mathbf{W}^{\mathbf{i}}_{\mathbf{AF}}}$, $\mathbf{b}^{\mathbf{i}}_{\mathbf{AF}} = \mathbf{b}^{\mathbf{i}}_{\mathbf{AF}} - \eta_{AF}\frac{\partial Loss_{AF}}{\partial \mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}}$
  if $Loss_{AF}$ convergence **then**
    break loop
  end if
  $j = j + 1$
end for
return $\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}}$ and $\mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}$
Initialize $\mathbf{Z}^0_{\mathbf{A}B}$
for $k = 1; k \leq I$ **do**
  $\mathbf{Z}^{\mathbf{i}}_{\mathbf{AB}} = g_i\left(\mathbf{W}^{\mathbf{i}}_{\mathbf{AF}} \cdot \mathbf{Z}^{\mathbf{i}-1}_{\mathbf{AB}} + \mathbf{b}^{\mathbf{i}}_{\mathbf{AF}}\right)(i = 1, \cdots, N)$
  $16: \mathbf{Z}^{N}_{\mathbf{AB}} = g_N\left(\mathbf{W}^{N}_{\mathbf{AF}} \cdot \mathbf{Z}^{\mathbf{N\text{-}1}}_{\mathbf{AB}} + \mathbf{b}^{N}_{\mathbf{AF}}\right)$
  $Loss_{AB} = MSE(\mathbf{Z}^{N}_{\mathbf{AB}}, [\sigma^L_{HH}, \sigma^L_{VV}])$
  $\mathbf{Z}^0_{\mathbf{AB}} = \mathbf{Z}^0_{\mathbf{AB}} - \eta_{AB}\frac{\partial Loss_{AB}}{\partial \mathbf{Z}^0_{\mathbf{AB}}}$
  if $Loss_{AB}$ convergence **then**
    break loop
  end if
  $k = k + 1$
end for
return $\mathbf{Z}^0_{\mathbf{A}B}$

Output: Inversion results $\mathbf{Z}^0_{\mathbf{A}B}$

References

1. Mohammad, H.M.; Amir, A.; Hamid, S.S. Substitution of satellite-based land surface temperature defective data using GSP method. *Adv. Space Res.* **2021**, *67*, 3106–3124.
2. Kim, Y.; Jackson, T.; Bindlish, R.; Lee, H.; Hong, S. Monitoring soybean growth using L-, C- and X-band scatterometer data. *Int. J. Remote Sens.* **2013**, *34*, 4069–4082. [CrossRef]
3. Yang, H.; Guo, H.D.; Wang, C.L.; Li, X.W.; Yue, H.Y. Polarimetric SAR surface parameters inversion based on network. *J. Remote Sens.* **2002**, *6*, 451–455.
4. Shen, X.; Mao, K.; Qin, Q.; Hong, Y.; Zhang, G. Bare surface soil moisture estimation using double-angle and dual-polarization L-band radar data. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 3931–3942. [CrossRef]
5. Chiang, C.Y.; Chen, K.S.; Yang, Y.; Wang, S.Y. Computation of backscattered fields in polarimetric SAR imaging simulation of complex targets. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 2004113. [CrossRef]
6. Sancer, M. Modified Beckmann-Kirchhoff scattering model for rough surface with large incident and scattering angles. *Opt. Eng.* **2007**, *46*, 078002.
7. Thorsos, E.I. The validity of the perturbation approximation for rough surface scattering using a Gaussian roughness spectrum. *Acoust. Soc. Am.* **1989**, *86*, 261–277. [CrossRef]
8. Soto-Crespo, J.M.; VesPerinas, M.N.; Friberg, A.T. Scattering from slightly rough random surfaces: A detailed study on the validity of the small perturbation method. *J. Opt. Soc. Am. A* **1990**, *7*, 1185–12017. [CrossRef]
9. Gilbert, M.S.; Johnson, M.S. A study of the higher-order small-slope approximation for scattering from a Gaussian rough surface. *Waves Random Media* **2003**, *13*, 137–149. [CrossRef]
10. Berginc, G.; Bourrely, C. The small-slope approximation method applied to a three-dimensional slab with rough boundaries. *Prog. Electromagn. Res.* **2007**, *73*, 131–211. [CrossRef]
11. Xu, F.; Jin, Y.Q. Imaging simulation of po-larimetric SAR for a comprehensive terrain scene using the mapping and projection algorithm. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 3219–3234. [CrossRef]

12. Zeng, J.Y.; Chen, K.S.; Bi, H.Y.; Zhao, T.J.; Yang, X.F. A comprehensive analysis of rough soil surface scattering and emission predicted by AIEM with comparison to numerical simulations and experimental measurements. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 1696–1708. [CrossRef]
13. Chen, K.S.; Wu, T.D.; Tsang, L.; Li, Q.; Shi, J.; Fung, A.K. Emission of rough surfaces calculated by the integral equation method with comparison to three-dimensional moment method simulations. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 90–101. [CrossRef]
14. Ulaby, F.T.; Sarabandi, K.; Mcdonald, K.Y.L.E.; Whitt, M.; Dobson, M.C. Michigan microwave canopy scattering model. *Int. J. Remote Sens.* **1990**, *11*, 1223–1253. [CrossRef]
15. Dubois, P.; Van Zyl, J.; Engman, T. Measuring soil moisture with imaging radars. *IEEE Trans. Geosci. Remote Sens.* **1995**, *33*, 915–926. [CrossRef]
16. Oh, Y. Quantitative retrieval of soil moisture content and surface roughness from multipolarized radar observations of bere soil surfaces. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 596–601. [CrossRef]
17. Zhao, S.T. Inverse calculation of hydrogeological parameters in Henan based on improved genetic algorithm. *Ground Water* **2019**, *41*, 77–79.
18. Wang, L.X.; Wang, A.Q.; Huan, Z.X. Parameter inversion of rough surface optimization based on multiple algorithms for SVM. *Chin. J. Comput. Phys.* **2019**, *36*, 577–585.
19. Peurifoy, J.; Shen, Y.; Jing, L.; Yang, Y.; Cano-Renteria, F.; DeLacy, B.G.; Joannopoulos, J.D.; Tegmark, M.; Soljačić, M. Nanophotonic particle simulation and inverse design using artificial neural networks. *Sci. Adv.* **2018**, *4*, eaar4206. [CrossRef]
20. Li, G.; Li, X.Z.; Liu, D.J.; Wang, L.H.; Yu, Z.F. A bidirectional deep neural network for accurate silicon color design. *Adv. Mater.* **2019**, *31*, 1905467.
21. Xu, F.; Wang, H.P.; Jin, Y.Q. Deep learning as applied in SAR target recognition and terrain classification. *J. Radars* **2017**, *6*, 136–148.
22. Sharifzadeh, F.; Akbarizadeh, G.; Kavian, Y.S. Ship classification in SAR images using a new hybrid CNN-MLP classifier. *J. Indian Soc. Remote Sens.* **2019**, *47*, 551–562. [CrossRef]
23. Ding, J.; Chen, B.; Liu, H.W.; Huang, M.Y. Convolutional Neural Network with Data Augmentation for SAR Target Recognition. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 364–368. [CrossRef]
24. Niu, S.R.; Qiu, X.L.; Lei, B.; Ding, C.B.; Fu, K. Parameter extraction based on deep neural network for SAR target simulation. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4901–4914. [CrossRef]
25. Oh, Y.; Sarabandi, K.; Ulaby, F.T. An empirical model and inversion technique for radar scattering from bare soil surfaces. *IEEE Trans. Geosci. Remote Sens.* **1992**, *30*, 370–381. [CrossRef]
26. Yang, Y.; Chen, K.S.; Shang, G.F. Surface parameters retrieval from fully bistatic radar scattering data. *Remote Sens.* **2019**, *11*, 596. [CrossRef]
27. Chen, K.S. *Radar Scattering and Imaging of Rough Surfaces*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2021; pp. 160–163.
28. Yang, Y.; Chen, K.S.; Tsang, L.; Yu, L. Depolarized backscattering of rough surface by AIEM model. *IEEE J. Sel. Top. Appl. Earth Sci. Remote Sens.* **2017**, *10*, 4740–4752. [CrossRef]
29. Zhang, Y.Y.; Wu, Z.S.; Zhang, Y.S. The effective permittivity and roughness parameters inversion by the land backscattering measured data. *Chin. J. Radio Sci.* **2016**, *31*, 79–84.
30. So, S.; Badloe, T.; Noh, J.; Bravo-Abad, J.; Rho, J. Deep learning enable inverse design in nanophotonics. *Nanophotonics* **2020**, *9*, 1041–1057. [CrossRef]
31. Li, J.; Li, X.Z.; Wu, Q.X.; Wang, L.H.; Li, G. Neural network enabled metasurface design for phase manipulation. *Opt. Express* **2021**, *29*, 2521–2528.

remote sensing

MDPI

Article

Object Tracking and Geo-Localization from Street Images

Daniel Wilson [1,†], Thayer Alshaabi [1,†], Colin Van Oort [1,†], Xiaohan Zhang [1], Jonathan Nelson [2] and Safwan Wshah [1,*]

[1] Complex Systems Center, University of Vermont, 194 South Prospect Street Burlington, Burlington, VT 05405, USA; daniel.wilson@uvm.edu (D.W.); thayer.alshaabi@uvm.edu (T.A.); cvanoort@uvm.edu (C.V.O.); xiaohan.zhang@uvm.edu (X.Z.)
[2] Penn State Department of Geography, 302 N Burrowes Street, University Park, PA 16802, USA; jkn128@psu.edu
* Correspondence: safwan.wshah@uvm.edu
† These authors contributed equally to this work.

Abstract: Object geo-localization from images is crucial to many applications such as land surveying, self-driving, and asset management. Current visual object geo-localization algorithms suffer from hardware limitations and impractical assumptions limiting their usability in real-world applications. Most of the current methods assume object sparsity, the presence of objects in at least two frames, and most importantly they only support a single class of objects. In this paper, we present a novel two-stage technique that detects and geo-localizes dense, multi-class objects such as traffic signs from street videos. Our algorithm is able to handle low frame rate inputs in which objects might be missing in one or more frames. We propose a detector that is not only able to detect objects in images, but also predicts a positional offset for each object relative to the camera GPS location. We also propose a novel tracker algorithm that is able to track a large number of multi-class objects. Many current geo-localization datasets require specialized hardware, suffer from idealized assumptions not representative of reality, and are often not publicly available. In this paper, we propose a public dataset called ARTSv2, which is an extension of ARTS dataset that covers a diverse set of roads in widely varying environments to ensure it is representative of real-world scenarios. Our dataset will both support future research and provide a crucial benchmark for the field.

Keywords: deep learning; object geo-localization; object detection; object tracking; traffic sign dataset

Citation: Wilson, D.; Alshaabi, T.; Van Oort, C.; Zhang, X.; Nelson, J.; Wshah, S. Object Tracking and Geo-Localization from Street Images. *Remote Sens.* **2022**, *14*, 2575. https://doi.org/10.3390/rs14112575

Academic Editors: Yue Wu, Kai Qin, Qiguang Miao and Maoguo Gong

Received: 6 April 2022
Accepted: 21 May 2022
Published: 27 May 2022

1. Introduction

Due to the rise of the internet and social media platforms, there exists an overwhelming quantity of publicly available images containing key geospatial information in the background. Furthermore, most modern hardware automatically records the location at which an image was taken. Most notably, transportation departments collect millions of street images every year. The purpose of these images is to manage road assets for road safety purposes; therefore, recognizing and geo-localizing road assets from these images is of extreme importance to many applications.

Object geo-localization is the process of taking objects identified in one or more images and determining their geospatial location represented as global positioning system (GPS) coordinates. It has a variety of applications including land surveying, self-driving vehicles [1], asset management [1–3], and any other domain that might benefit from the capability to automatically detect and geolocate objects of interest [4,5].

Determining objects' GPS locations from street images can be a cheap solution for road asset geo-localization, but this task is also very challenging due to GPS error, multiple appearances of the same objects in images or frames, the variety of object types (for example road signs can contain more than 200 sub-classes), etc. A particularly challenging component of this problem is the lack of a pre-defined relationship between the number of

images and how many times each object appears in the dataset. Objects may appear in one, two, or any number of images, meaning an algorithm must both detect re-occurrences of the same object across multiple images, and then collapse them into a single prediction.

Geo-localization algorithms can be categorized based on how they handle repeated detections. Triangulation-based methods use a classic triangulation approach to determine an object's GPS using the depth to an object in an image and the image's coordinates, and typically a clustering algorithm to condense repeated object occurrences [6,7]. Re-identification approaches use an object detector that detects objects by receiving multiple frames as input, which implicitly merges repeated detections across the multiple input frames [2,3]. Tracker-based approaches separately detect objects in each frame, and re-occurrences of objects across multiple frames are identified using an object tracking algorithm [1].

Ref. [6] proposed a triangulation-based method using a two-stage framework which performed object segmentation and then object geo-localization. They later improved their approach in [7] by combining footage from a drone point cloud to enhance accuracy. These methods have an inherent performance ceiling as they rely on noisy segmented objects. In addition, they assume object sparsity, in which all objects within a certain distance threshold are assumed to be a single object.

Re-identification methods were proposed by [2]. Their model receives two images as input, and jointly detects and geo-localizes objects between those two frames. Following this idea, they proposed a graph-based approach in [3] to handle greater than two frames. A limitation is these methods require the objects to appear in at least two frames. In addition, they assume all objects are close to the camera for easy detection.

Recently, ref. [1] proposed a tracking-based method to geo-localize traffic signs using a deep neural network that was mostly end-to-end trainable. Their architecture detects objects, predicts their pose in five dimensional space, and then associates those objects between frames. In their approach, they only selected objects appearing in at least five frames. Their system required a total of six cameras, imposing a crucial hardware limitation.

In addition to the aforementioned drawbacks of each technique, most notably, all share a major limitation in which they are only capable of geo-localizing one class of objects. In addition, many of these approaches rely on expensive or uncommon hardware not accessible in many use cases. For example, ref. [7] relies on drone footage, and ref. [1] uses an array of six cameras, which requires the use of specialized hardware.

An additional current pitfall in the field is the use of datasets constructed exclusively in a single environment, such as city streets [1,2]. Datasets also commonly only annotate occurrences of objects close to the camera, since these are the easiest for an algorithm to detect [1–3]. Datasets also contain objects that are visually distinct and spaced far apart from one another [1–3], making them easier to distinguish. A comprehensive survey of the field of object geo-localization is provided in [8].

In this paper, we seek to rectify the limitations of the current algorithms by proposing a new tracking-based deep learning approach to geo-localize dense objects from low frame rate video using a novel tracker algorithm. The proposed approach handles multi-class objects that might exist in one or multiple frames and uses only cheap hardware. Our proposed system relies exclusively on a single camera, each image's GPS location, and the image heading, which makes our system practical for mass adoption.

We also propose a new dataset for benchmarking geo-localization algorithms. Our proposed dataset is an extension of [9]. We capture a variety of driving environments, and achieve a broad class distribution containing 199 different sign types. Crucially, it contains clusters of signs with similar and in some cases identical appearance, posing a very challenging and much more realistic benchmark compared to previous datasets.

Our proposed dataset and methodology is not limited to traffic signs. Our system is generalizable and could easily applied to other applications including geo-localization of telegraph poles, painted street markings, traffic lights, side walks, trees, buildings, and any other land features of interest. Our dataset provides a crucial benchmark that any

class-based geo-localization algorithm from these domains could use as an additional benchmark to aid in research and development.

Our research contributions can be summarized as:

1. An enhanced version of the ARTS [9] dataset, ARTSv2, to serve as a benchmark for the field of object geo-localization.
2. A novel object geo-localization technique that handles a large number of classes and objects existing in an arbitrary number of frames using only accessible hardware.
3. An object tracking system to collapse a set of detections in a noisy, low-frame rate environment into final geo-localized object predictions.

2. Related Work

A somewhat similar area of research are simultaneous localization and mapping (SLAM) algorithms which are designed to model the surrounding environment typically for the purposes for vehicle navigation [10]. By contrast, the purpose of geo-localization algorithms is to determine object positions on a global scale by predicting their GPS coordinates and building a geographical information systems (GIS) map. Furthermore, since SLAM is intended primarily for navigation, these algorithms are designed to run in real time. Object geo-localization algorithms can be applied to pre-existing datasets, since they are not necessarily intended for real time applications.

Object geo-localization from images has been the focus of important recent research. Before deep learning, the most common approach for object geo-localization was to use epipolar constraints [11] to reconstruct 3D points from corresponding image locations. This method has been used to predict traffic light locations [12], and to triangulate and estimate the locations of traffic signs that were detected from their silhouette [13]. A related approach [14] proposed a pipeline that triangulated telecom assets using a histogram of oriented gradients (HOG) as feature descriptors, along with a linear SVM [15] from Google Street View (GSV) images. These methods suffer from poor performance as they used handcrafted features.

Deep neural networks (DNNs) have become the new state-of-the-art technique in geo-localization due to their capabilities to capture complex relationships directly from data through building an effective hierarchical feature representation. While it is already common practice to detect objects in images using deep learning approaches, object geo-localization has the additional requirement that objects appearing in multiple images must be merged into a single prediction. There are three core approaches to accomplishing this merging. First, in triangulation-based approaches, triangulation is used to determine object geo-locations and then a clustering algorithm is typically employed to merge repeated detections [6,7]. The second class of approaches are re-identification-based. In these approaches, a model jointly detects objects using multiple frames as input. When making predictions, these models produce a single prediction for an object from the multiple input frames, thus implicitly merging objects in those frames into a single prediction [2,3]. Third, tracker-based approaches explicitly associate objects between frames, forming tracklets of detections from the same object [1]. These tracklets can then be condensed using a weighted average or a similar approach to create a final sign prediction.

The first triangulation-based approach was proposed by [6], who built a framework that uses a convolutional neural network (CNN) to perform monocular depth estimation from images. They used a Markov random field (MRF) to triangulate the coordinates of the detected objects, and merged the repeated occurrences of objects across multiple images using a clustering algorithm. The authors later expanded their method by incorporating point cloud data captured from drones to enhance geo-localization accuracy [7]. This enhancement came at the cost of introducing a hardware constraint due to requiring drone footage. Triangulation methods are limited in their performance as they rely on noisy segmented objects. Ref. [16] proposed to reduce the noise associated with this method using a structure from motion technique. All these approaches contain the fundamental

assumption of object sparsity, in which all objects within a certain distance threshold are assumed to be a single object.

The first re-identification-based method was proposed by [2], who combined object detection and re-identification into a joint learning task using a soft geometric constraint on detected objects from GSV images. The largest limitation of this approach is it required each object to appear in exactly two images, which was not a reasonable real world assumption. To address this limitation the same authors [3] proposed GeoGraph, a graph neural network (GNN)-based method for geo-localization, which is capable of jointly detecting objects in more than two frames. Both these models require a fixed number of input images to be determined before training. Real-world data do not contain objects that disappear after a fixed number of frames, meaning these approaches are not sufficient for real scenarios.

The only tracker-based approach was proposed by [1]. They constructed a deep neural network consisting of an object pose regression network and an object matching network. The object pose regression network detects objects and predicts their 5D pose. The object matching network matches the detected objects to combine objects with repeated appearances in multiple images. The limitation of this approach is that the camera's intrinsic matrix along with six different image perspectives were used as input to the algorithm, meaning specialized hardware must be used to gather the inputs for the model.

In addition to the drawbacks mentioned for these techniques, most notably, all share a major limitation in which they are only capable of geo-localizing one class of objects. In this paper, we are going to propose a new multi-class tracking-based technique for object geo-localization from images. Our proposed technique can handle objects that might exist in one or multiple frames. Our algorithm uses a single camera, the image's GPS location, and the image's heading, which makes our system viable for mass adoption using a cheap hardware.

Many general purpose tracking-by-detection frameworks have been developed over the past decade for a wide range of applications [17–22]. The most common approach is to use visual cues and motion tracking to trace objects in a sequence of images [23–26]. An alternative approach is to train a model to explicitly measure the similarity of each pair of objects. Ref. [27] constructed a deep siamese convolutional network to learn such a similarity function, which was trained during an offline learning phase and then evaluated during tracking. Another approach is to model multiple object tracking using a Markov decision process (MDP), as proposed by [28]. A final noteworthy approach uses dual matching attention networks to incorporate both spatial and temporal information [23]. The networks generate attention maps on input images, which are used to perform tracking.

Most object geo-localization datasets are limited to low frame rates. Traditional object trackers are designed for high frame rate data in which objects only move small distances between frames. They cannot be effectively applied to datasets where there are large jumps between frames. Furthermore, traditional trackers are not designed to take advantage of objects' GPS coordinates as additional information with which to perform association between frames. We therefore cannot apply traditional object tracking approaches to our dataset, and instead opt to design a novel tracker to address the unique properties of our geo-localization dataset.

3. Datasets

3.1. Existing Datasets

Despite recent interest, only a limited of datasets have been proposed to support research in object geo-localization. There are three major datasets (Pasadena, TLG, and ARTS) which are summarized in Table 1.

Ref. [2] proposed a multi-view dataset in which the goal is to re-identify multiple occurrences of street side trees from different views. It includes 6020 individual trees, 6141 GSV images formatted as panoramas, and 25,061 bounding boxes. Each tree was annotated from its four closest panoramas, and is labeled with a unique ID so re-identification can be performed; however, their dataset is not publicly available, and is limited due to not con-

taining distinct classes of objects. It is limited in its size due to only containing 6141 images. This dataset also assumes object sparsity, meaning that all objects within a nearby radius are assumed to be the same object. Furthermore, their dataset does not contain clusters of objects, which is the most challenging scenario for object geo-localization algorithms.

Table 1. A comparison between the Pasadena multi-view object re-identification [2], the traffic light geo-localization (TLG) [1], ARTS v1.0 easy and challenging [9], and ARTSv2.0 datasets.

	Pasadena Multi-View ReID [2]	**Traffic Light Geo-Localization (TLG) [1]**	**ARTS v1.0 [9]**		**ARTSv2.0**
			Easy	**Challenging**	
Number of classes	1	1	78	171	199
Number of images	6141	96,960	9647	19,908	25,544
Number of annotations	25,061	Unknown	16,540	35,970	47,589
Side of the road					✓
Assembly					✓
Unique Object IDs	✓	✓			✓
5D Poses		✓			
GPS	✓	✓	✓	✓	✓
Color Channels	RGB	RGB	RGB	RGB	RGB
Image Resolution	2048 × 1024	1600 × 1900	1920 × 1080	1920 × 1080	1920 × 1080
Publicly Available		✓	✓	✓	✓

Researchers from Uber [1] compiled another dataset for traffic light detection derived from nuScenes, a popular open-source dataset for autonomous driving [29]. The dataset has 400 scenes, each lasting 20 s with 12 frames per second. All images have metadata indicating the 5D pose of the camera and each annotated traffic light. Each traffic light can be distinguished by a uniquely assigned ID. Their dataset is also limited in that it lacks object classes. It is built from images in a single city-like environment, which lacks the variation associated with data from the real world. Objects are only selected for the dataset if they appear in at least five keyframes. These assumptions artificially reduce the difficulty of the dataset relative to the real world. This dataset is also reliant on the availability of the camera's intrinsic matrix, which requires the use of specialized hardware to capture.

The third noteworthy dataset was ARTS proposed by [9]. The original ARTS dataset is composed of nearly 20,000 images containing 171 different classes of signs. The dataset is structured as sequences of images referred to as road segments. Each segment contains a sequence of images taken from a camera mounted to the top of a car driving down a road, with roughly one second intervals between each image to satisfy storage constraints. Each image contains an annotation for each readable sign, and each annotation specifies a bounding box around the sign, the sign's class, and the GPS coordinates of that sign. The camera's coordinates and heading are also available for each image. The ARTS dataset contains an easy and challenging subset, along with a third format referred to as video logs. All three configurations of the dataset provide manually labeled annotations in a format similar to PASCAL VOC [30]. The easy version of the dataset contains a total of ∼10 K images and ∼17 K annotations, covering 78 different sign classes. All annotated signs in the easy version were captured at up to a 100 m radius of the camera with a minimum of 50 samples per class. The challenging version of the dataset contains a total of ∼35 K annotations scattered in ∼20 K images, covering 171 sign classes, with a minimum of 20 samples per class captured from a distance up to 100 ms. The video logs contain the raw sequences of images and their annotations in the same directory, without being organized into train, validation, and test sets.

The ARTS dataset would benefit from more training samples to address its sparse class distribution by providing more effective samples per class. In addition, a limitation of the ARTS dataset is that it lacks unique identifiers to indicate repeated occurrences of the same sign in multiple images, which inhibits the capability of researchers to benchmark models on this dataset. In the following section, we are going to propose our extension to the ARTS dataset, which will be the largest dataset for traffic sign geo-localization and benchmarking.

3.2. ARTSv2 Dataset

Substantial enhancements have been made to the ARTS dataset [9] to construct ARTSv2. We have increased the number of images to 25,544, the number of unique sign classes to 199, and the number of annotations to 47,589. These enhancements help provide more training samples for less common sign classes, which is one of the fundamental problems with this dataset. Moreover, each sign annotation has been updated with additional attributes. First, each annotation specifies the 'sign side', which indicates the side of the road the sign is on, represented as a string indicating left, right, or other. The "other" string is provided for signs that should not be labeled as either left or right, such as signs attached overhangs above the road. Second, each annotation has a binary attribute marking whether the sign is part of an assembly. A sign assembly refers to a group of signs supported by the same post. An example assembly is shown in Figure 1. Each sign that is part of an assembly will have this boolean attribute annotated as True, whereas stand-alone signs that are not part of an assembly will have this attribute set to false. Finally, each physical sign in a road segment has been given unique integer identifier. Since most signs appear in multiple images, a sign annotation will have the same ID each time the that physical sign appears. These unique identifiers are crucial since in order to evaluate the performance of geo-localization algorithms, repeated occurrences of the same object must be identified. Sample images are shown in Figure 2.

All the systems proposed and implemented in this paper use the ARTSv2 dataset.

Figure 1. An example of a sign assembly containing multiple signs of similar appearance. All of the signs on the assembly have a green and white appearance, so it is difficult for a model to distinguish between them. There are two signs containing the word "East" which appear essentially identical. There are also two signs with the text "Vermont 15". The arrow in the bottom left is a mirrored version of the arrow in the middle right. Since these signs contain so many similar characteristics, and in some cases are nearly identical, it is extremely challenging to create a geo-localization model that separately geo-localizes these signs.

Figure 2. Sample images from the ARTSv2 dataset. The images contain a variety of sign types, often clustered very close together, which makes for challenging geo-localization. Environment and road types also vary widely.

3.3. Dataset Construction

To construct this dataset, images were first gathered from a vehicle with a top-mounted camera, which records footage while traveling in the State of Vermont in the United States. The vehicle travels across the state to capture footage in a wide range of environments, including highways, cities, and rural streets. Since the storage constraints associated with storing so much video would be prohibitive, frames along with their respective GPS and headings are extracted from the video at approximately 1 s intervals. To construct the annotated dataset from these images, human annotators used a version of labelImg [31], which we have modified with the capability to annotate each sign's GPS coordinates, road side, assembly attribute, and unique integer identifier. This tool will be made publicly available to support the construction of other geo-localization datasets. An image of the user interface is shown in Figure 3.

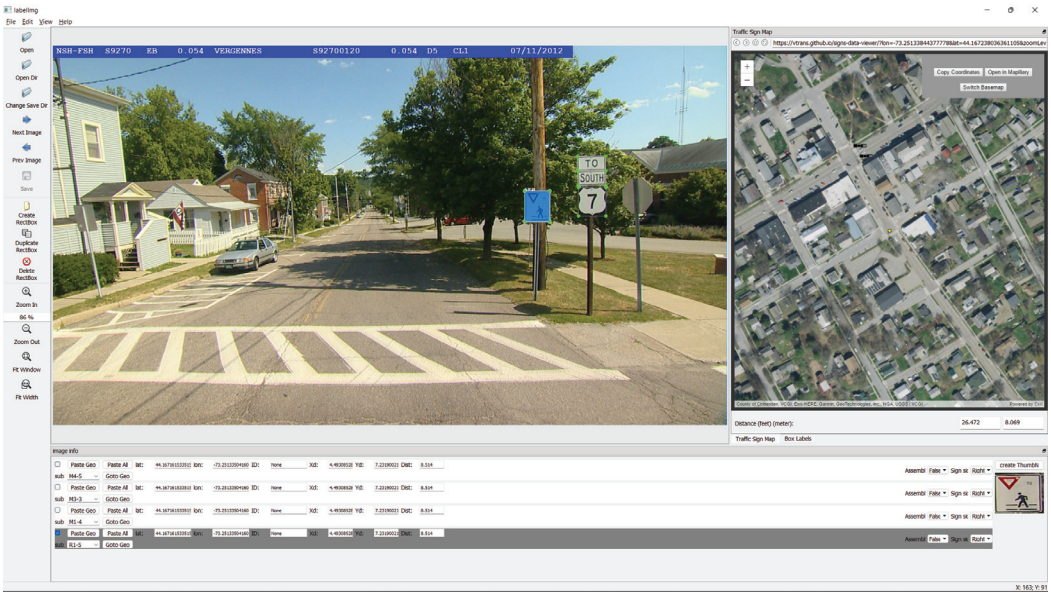


Figure 3. A sample image of the graphical user interface provided by our modified version on labelImg. The image being annotated is displayed in the upper left, and bounding boxes that have been annotated are overlaid on the image. The user can use the map displayed in the upper right to select the GPS coordinates for each sign. In the bottom pane, the user can enter all appropriate information associated with the sign, including its class, GPS coordinates, assembly attribute, sign side, and integer identifier.

3.4. Unique Characteristics

Compared to other traffic recognition and geo-localization datasets, ARTS is the largest in terms of both the number of images, classes, and annotations. The dataset contains high quality 1920 × 1080 resolution images, available in multiple formats including video logs and individual annotations in a format similar to the PASCAL VOC format. ARTSv2 is also the only dataset containing labels specifying side of road and assembly attributes. Table 1 shows a full comparison between ARTS and similar geo-localization datasets in terms of number of classes, number of images, and number of annotations for each dataset.

Current datasets for object geo-localization algorithms are simple and constructed under ideal circumstances [1,2]. The ARTSv2 dataset contains multiple unique challenges, which makes it more representative of circumstances encountered in the real world. First, ARTSv2 features 199 different sign classes appearing with a highly imbalanced distribution, thereby classes such as stop signs appear far more frequently than more obscure classes of signs. This is an important characteristic of our dataset, since imbalanced class distributions are a substantial challenge currently faced by machine learning models. The heavy-tailed distribution increases the difficulty of training models to predict sign classes appearing less frequently because they have fewer training samples. In addition to posing a significant challenge, this class imbalance is much more representative of what we expect to see in the real world compared to other datasets. This class imbalance is visually illustrated in Figure 4, which shows the cumulative probability distribution of class frequencies in the dataset.

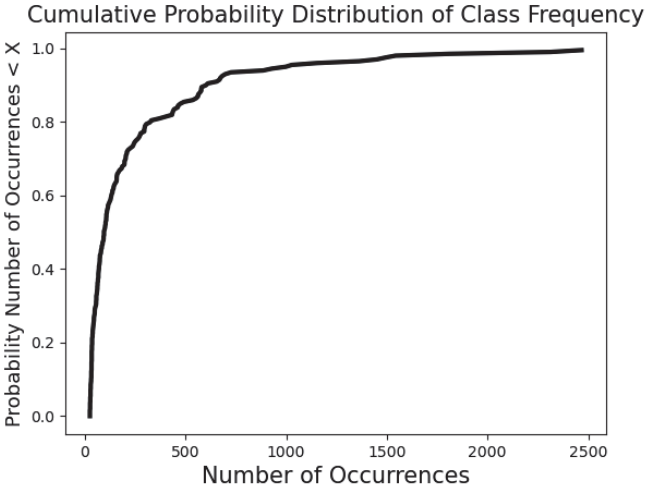


Figure 4. A cumulative class distribution plot showing the distribution of frequencies at which different classes appear in ARTSv2. The *x*-axis indicates a class frequency, and the *y*-axis value indicates the probability that a class occurs at most the number of times indicated on the *x*-axis. The sharp rise on the left side of the graph shows there are many classes that appear with low frequency, posing a unique challenge for geo-localization algorithms, which must adapt to classes with few training samples.

US traffic sign classification also faces the unique challenge of inconsistency between states. While the US Department of Transportation standards are followed to varying degrees, there are a wide variety of specific traffic sign configurations across state road networks. Roads contain many signs that do not conform to known standards. Classifying these non-standard signs therefore poses another unique challenge, as models must learn to cope with signs that may be truly unique, meaning that they only appear once in the entire dataset. Signs that do not fit into a clear category were annotated with an "unknown" class label.

Another unique challenge associated with this dataset is the existence of many objects with similar appearances to road signs, which tends to create false positives from object detectors. Business signs and billboards, hand-made signs placed for events such as yard sales, and car license plates tend to create false positives because they contain visual characteristics similar to road signs. Models trained on this dataset therefore face the challenge of learning to distinguish between road signs, sign-like objects, and other signs that are not technically classified as road signs.

The ARTSv2 dataset was captured in a wide variety of driving environments. There are road segments corresponding to highways, small rural roads, complex intersections, and busy city roads. The vehicle travels at a variety of speeds, takes many turns, and moves up and down hills, which causes signs to change their positions unpredictably between frames. The vehicle may move between other cars or trees such that a sign is visible in one frame and obscured in the next, only to re-appear again a few frames later. Unlike other datasets, we do not remove these non-ideal scenarios since we expect them to be encountered when applying this technology to the real world.

Finally, sign assemblies are a particularly challenging component of our dataset for several reasons. First, assemblies contain clusters of nearby signs that need to be individually detected and geo-localized. Clusters of nearby objects is the most common challenge for object geo-localization algorithms, which is why other datasets have opted to remove them. This challenge is compounded by the fact that signs of similar appearance are partic-

ularly likely to occur on the same assembly, since assemblies tend to group together signs intended for a specific function, such as an indicating nearby highways. These assemblies of signs have similar GPS coordinates and often extremely similar appearances, meaning there are few features a model can use to distinguish between these objects. Clusters of similar objects is the most difficult characteristic of the ARTSv2 dataset, which is a challenge that has been neglected by previous research.

4. Materials and Methods

4.1. System Overview

At a high level, our system is composed of two core stages as displayed in Figure 5.

In the first stage, road images are provided to a modified RetinaNet we have constructed called GPS-RetinaNet. GPS-RetinaNet receives these images as input, and outputs a bounding box around each sign, its sign class, and its geospatial location. Since most signs will appear in multiple images, the purpose of the second stage of our system is to condense these repeated detections into a single prediction. First, we train a similarity network, which receives pairs of sign detections predicted by GPS-RetinaNet as input. The similarity network learns to predict a scalar value that measures how similar its input detections are. Next, we used a modified variant of the Hungarian algorithm to pair detections of high similarity. Intuitively, detections with high similarity are more likely to be from the same sign. A list of signs paired together by the Hungarian algorithm is referred to as a tracklet. Each tracklet is then condensed into final sign prediction using a weighted average, producing the final GIS map as shown in Figure 6. The following sections will break the components of this pipeline down in more detail.

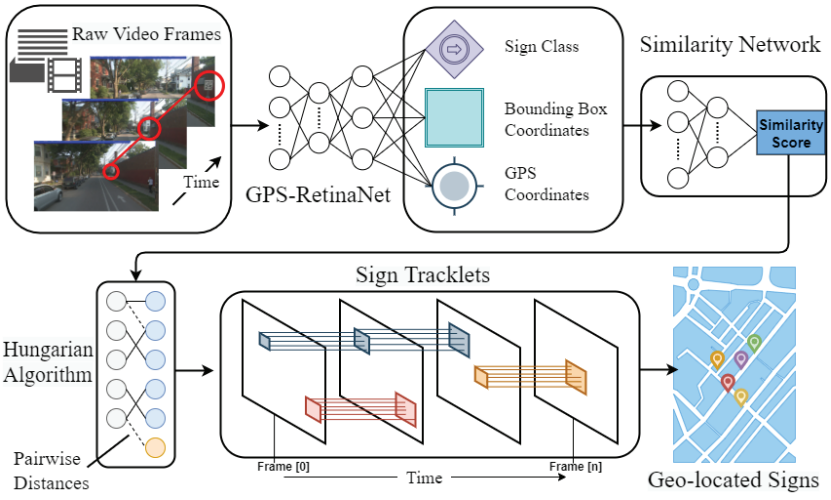


Figure 5. An overview of the Sign Hunter pipeline. First, raw images extracted from videos of a road vehicle (top-left) are fed into GPS-RetinaNet (top-middle) which detects, classifies, and predicts signs' GPS offsets. Pairs of detections output by GPS-RetinaNet are provided as input to the similarity network (top-right), which quantifies the similarity between the signs. The Hungarian algorithm [32] (bottom-left) uses the similarity scores to merge repeated occurrences of objects, which forms tracklets (bottom-middle) containing all the occurrences of each object in the dataset. These tracklets are condensed into final sign predictions to create a GIS map (bottom-right) of sign locations.

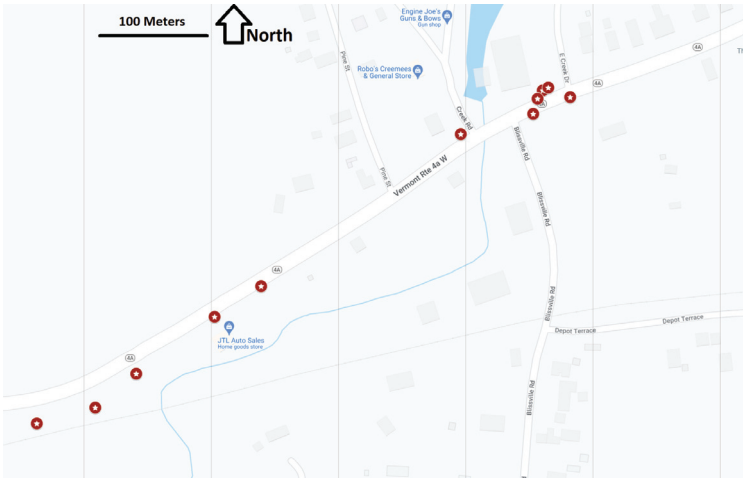


Figure 6. The end result of our pipeline is that signs are classified based on their sign type and placed on a map corresponding to their geo-location. Each dot indicates the location of a single sign after all the repeated detections have been merged. Properties of the sign such as its class can be inspected by clicking on it on the map tool.

4.2. GPS RetinaNet

The first stage of our model performs three functions. It detects each sign visible in a road side image, classifies what type of sign it is, and regresses its geospatial coordinates. We have constructed a system with these capabilities by modifying the popular object detector RetinaNet [21]. RetinaNet is an object detector already capable of performing detection and classification. It uses a backbone network as the core of the architecture, and employs a feature pyramid network to extract features from this backbone. The outputs from the feature pyramid are provided as input to two sub-networks, one of which regresses bounding boxes around objects and the other of which predicts the detected object's class. RetinaNet is not, however, capable of regressing geo-coordinates. We modified its architecture by building GPS-RetinaNet, which contains an additional fully connected GPS sub-network. The GPS sub-network extracts features from RetinaNet's feature pyramid [21] and learns to regress a detected object's offset in a coordinate system local to the image. We call this additional sub-network the GPS subnet, which expands RetinaNet's base architecture as is displayed in Figure 7. Each sub-network is composed of four fully connected convolutional layers with ReLU activations. The classification sub-network terminates with $(K \times A)$ linear outputs, where A is the number of different anchors used in the network and K represents the number of classes. The box-regression sub-network ends with $(4 \times A)$ linear outputs to determine the relative position of the object [21]. The GPS sub-network concludes with $(2 \times A)$ linear outputs for each spatial level in the network. We use the popular ResNet [33] as the backbone for this architecture. Sample outputs are shown in Figure 8.

Since directly predicting GPS coordinates of signs is challenging without knowing which direction an image is facing, we instead train the GPS sub-network to predict offsets relative to the image, which we then convert to the sign's actual GPS coordinates. In more detail, the GPS-subnet learns to regress two local offset values, indicating the horizontal and vertical distance to the sign in meters, which represents the position of the detected

object from the perspective of the camera image. These offsets are then fed into a coordinate transform to generate the object's predicted GPS location as follows:

$$X_r = X_o \times \cos\theta + Y_o \times \sin\theta \tag{1}$$
$$Y_r = X_o \times \sin\theta - Y_o \times \cos\theta \tag{2}$$
$$O_{lat} = Y_r / 6378137 \tag{3}$$
$$O_{lon} = X_r / (6378137 \times \cos(\pi \times C_{lat}/180)) \tag{4}$$
$$P_{lat} = C_{lat} + O_{lat} \times 180/\pi \tag{5}$$
$$P_{lon} = C_{lon} + Olon \times 180/\pi \tag{6}$$

The variables X_o and Y_o represent the respective horizontal and vertical offsets predicted by the network from the perspective of the image in meters. We use θ to represent the camera's facing direction (measured with a compass), and C_{lat} and C_{lon} indicate the camera's latitude and longitude. Both X_r and Y_r are calculated as the meter offsets along the longitudinal and latitudinal axis after being rotated from the camera's coordinate system. Hence, O_{lat} and O_{lon} are offsets converted from meters to latitude and longitude, and P_{lat} and P_{lon} provide the final latitude and longitude prediction of the detected object after adding the predicted offset of the camera coordinates.

To provide supervision when training the network, we must be able to calculate the desired offsets from the annotated GPS coordinates. In other words, in addition to the capability of converting the offsets predicted by the network to GPS coordinates, we also require the ability to invert this transformation and convert the annotated GPS coordinates to offsets. This can be simply accomplished by re-arranging the above formulas as show below, in which all of the variables remain the same, except that P_{lat} and P_{lon} are replaced with A_{lat} and A_{lon}, which represent the annotated latitude and longitude of the sign, respectively.

$$O_{lat} = (A_{lat} - C_{lat}) \times \pi/180 \tag{7}$$
$$O_{lon} = (A_{lon} - C_{lon}) \times \pi/180 \tag{8}$$
$$X_r = O_{lon} \times (6378137 \times \cos(\pi \times C_{lat}/180)) \tag{9}$$
$$Y_r = O_{lat} \times 6378137 \tag{10}$$
$$X_o = X_r \times \cos\theta + Y_r \times \sin\theta \tag{11}$$
$$Y_o = X_r \times \sin\theta - Y_r \times \cos\theta \tag{12}$$

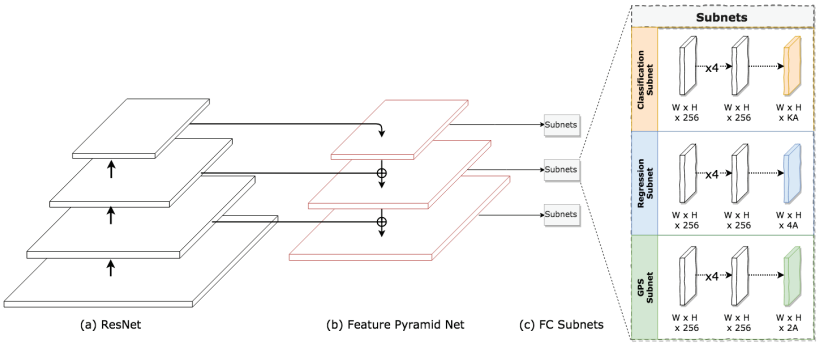


Figure 7. GPS-RetinaNet. Similar to RetinaNet [21], this architecture uses a FPN [17] backbone on top of a ResNet [33] model (**a**) to create a convolutional feature pyramid (**b**). Then, we attach three subnetworks (**c**); one for classification, one for box regression, and one for GPS/depth regression.

One of the most challenging characteristics of the ARTSv2 dataset is its heavy class imbalance. To address this, we propose a modification to Focal Loss [21] that replaces

γ in the original definition with an adaptive modulator. We define the new focusing parameter as:

$$\Gamma = e^{(1-p_t)}, \tag{13}$$

$$\mathrm{FL}e(p_t) = -(1-p_t)^{\Gamma}\log(p_t). \tag{14}$$

For convenience, we refer to our new definition of Focal Loss as (FL*e*) throughout the paper. FL*e* introduces two new properties to the original definition. First, it dynamically fine-tunes the exponent based on the given class performance to reduce the relative loss for well-classified classes while maintaining the primary benefit of the original FL. Figure A1 directly compares FL with FL*e*, highlighting that FL*e* (shown in green) crosses over $FL_{\gamma=2}$ (shown in orange) around ($p_t = 0.3$). As p_t goes up from $0.3 \rightarrow 1$, FL*e* starts to shift up slowly ranging in between FL and Cross Entropy CE (shown in blue). See Appendix A for more technical details. We use Fl*e* loss to train the classification sub-network, and we use the standard L1 loss to train the bounding box and GPS regression sub-networks.

Figure 8. Sample images and detections from the ARTSv2 datset. Images contain a variety of sign types, often clustered very close together, which makes for challenging geo-localization. Each box around each sign represents a separate detection from GPS-RetinaNet. The color of the box represents which tracklet the detection has been assigned to by the multi-object tracker. Since a tracklet is a list of signs predicted to be the same, re-occurrences of the same sign in multiple images should have the same color box around it.

4.3. Multi-Object Tracker

When GPS-RetinaNet is applied to an image, it produces detections for each sign specifying a bounding box, sign class, and (after a coordinate transform) GPS coordinates. Because images in the ARTSv2 dataset are taken approximately one second apart, the same sign will typically appear in multiple frames. Since our final goal is to produce one geo-localized sign prediction for each sign, we need to collapse the multiple detections

produced for many signs into a single prediction for each distinct, physical sign. Our proposed solution is a tracker that iteratively steps through the images in each road segment from the ARTSv2 video logs. As the tracker steps through the images, it merges repeated detections from the same signs appearing in multiple frames. This tracker is composed of two core components, the similarity network and the Hungarian algorithm. The role of the similarity network is to compute a learned heuristic indicating how likely it is a pair of detections provided by GPS-RetinaNet refer to the same sign. The second component, which is a modified variant of the Hungarian algorithm, uses these similarity scores as input to merge repeated detections. Our multi-object tracker is designed to operate in a low-frame rate environment in which objects can move considerable distances along unpredictable trajectories between frames due to the vehicle's motion. The tracker incorporates both the use of visual cues, predicted GPS position, predicted class, and relative bounding box position to address the core challenge posed by clusters of similar signs.

4.3.1. Similarity Network

To train the similarity network, we format the sign annotations from the ARTSv2 dataset to the same format as the detections output by GPS-RetinaNet, so that they can be used as the inputs when training the network. We can use the unique integer identifiers from ARTSv2 to determine if a pair of annotations fed to the similarity network are from the same sign, which will determine the appropriate output for the network during training.

As shown in Figure 9, the similarity network receives three types of inputs associated with each annotation. First, the similarity network receives a vector of values containing the image GPS, image heading, the sign class (represented as a 50 dimensional embedding vector), sign GPS, and bounding box. The second input to the similarity network is the pixels showing an image of the sign. The pixel information within the sign's bounding box is extracted and resized to a $32 \times 32 \times 3$ resolution using bi-linear interpolation. The third input to the network is a rank 3 tensor containing a "snapshot" encoding the spatial position of each sign relative to all other signs in the image. This is accomplished by assigning each sign in the frame to a square in a 10×10 grid, corresponding to its location in the image. The correct square to place the sign at is calculated using Formula (15), in which G_x and G_y represent the grid X and Y cells the sign is placed, B_x and B_y indicate the center coordinates of the sign's bounding box, H and W are the height and width of the image, and S is the size (in our case 10) of the grid. Along the depth axis at each square in grid containing a sign, we concatenate a vector containing that sign's GPS coordinates and its 50 dimensional class embedding. Grid locations that do not contain a sign are padded with a vector of zeros. The net result is a 3D tensor containing a "snapshot" of information encoding the relative position of signs in the image. This component of our architecture is crucial to address the challenge of signs with similar or identical appearance discussed in Figure 1. If two identical by appearance signs are in an image, this input can allow those signs to be distinguished based on their position in the grid relative to one another. Since the similarity network predicts the similarity of a pair of signs, it receives two instances of each of these three inputs, one of which is from each sign.

$$G_x = \lfloor B_x / W \rfloor \times S \tag{15}$$

$$G_y = \lfloor B_y / H \rfloor \times S \tag{16}$$

The network is trained to predict a value between 0 and 1, which represents the probability that a pair of detections belong to the same sign. These values can be interpreted as a similarity metric, where output values closer to 0 indicate the sign detections have greater similarity, and are thus less different from one another. Conversely, when receiving inputs from different signs the network should predict outputs closer to 1, indicating that the detections have less in common with one another.

The architecture of this network contains two siamese sub-networks and a third sub-network designed to handle the remaining inputs. The $32 \times 32 \times 3$ scaled images containing

the pixels from the detections are fed through the first siamese sub-network consisting of two sets of convolutional layers containing 32 3 × 3 convolutional filters followed by batch normalization. The resulting features are sent through two fully connected layers resulting in a vector of 32 features. The 3D tensors containing a snapshot of all the signs are processed by a second siamese sub-network consisting of two convolutional layers each containing 32 3 × 3 convolutional filters, which are followed by two fully connected layers resulting in a vector of 8 features. The fully connected outputs of these two siamese sub-networks are concatenated with the vector containing the camera heading and GPS, predicted sign GPS, sign class, bounding box, and sign class embedding. The resulting vector is sent through multiple fully connected layers to generate the final prediction. The exact architecture is shown in Figure 9.

The final task of training the similarity network is to develop a satisfactory noise distribution when training. During test time, the similarity network receives output detections from GPS-RetinaNet as input, but as discussed above we must train the similarity network on annotations from our dataset since they contain the labels indicating if two signs are the same. We therefore want to construct a noise distribution for these training annotations that mimics the noise introduced by our object detector. To find a noise distribution, we implemented an algorithm that tests if an annotation has an obvious detector output match when the image that annotations is from is provided as input to GPS-RetinaNet. First, we checked if the annotation's bounding box has one (and only one) detection in the same image for which their intersection over union (IOU) is greater than 0.9. If this is the case, we measure the latitude and longitude discrepancy between the annotation and detection, create a boolean variable indicating if the classes match, and subtract the differences between the X and Y coordinates of their bounding boxes. These three values quantify how much "noise" was introduced by the detector by measuring how different the annotation is from the corresponding detection predicted by GPS-RetinaNet. By repeating this process for each annotation, we construct a noise distribution representing how often and by how much the detected GPS, detected class, and detected bounding boxes differ from the annotated GPS, annotated class, and annotated bounding box. We can then stochastically sample from this noise distribution to serve as our data augmentation when training the similarity network.

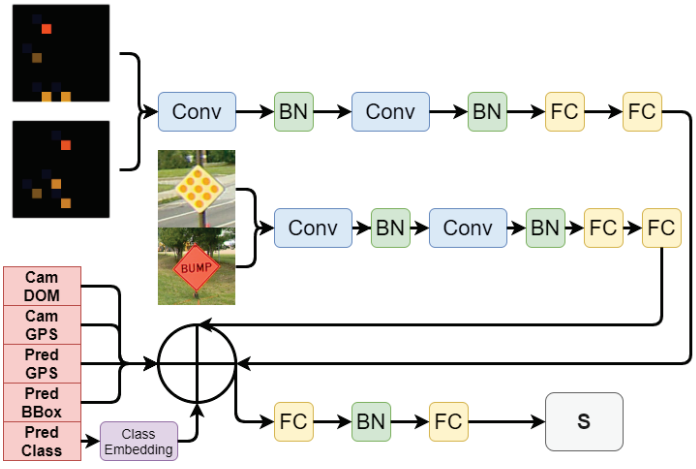


Figure 9. The architecture of the similarity network. The network uses two siamese sub-networks and then concatenates all the resulting features. The remaining features are sent through two more fully connected layers before predicting the similarity score.

We trained this network on an Nvidia GTX 1080 ti with 11 GB of VRAM and implemented the network using the Keras application programming interface with tensorflow

as the back-end. We regularized the inputs from each image such that each color value of each pixel was scaled from 0 to 1. Our network incorporated batch normalization after each layer, in addition to a dropout ratio of 0.25 after each fully connected layer. We fed inputs into the network in batch sizes of 128, and used the Adam optimizer [34] with a learning rate of 0.0001 to optimize the weights of the network. We trained the network for 20 epochs. Finally, since we found categorical cross entropy led to poor performance, we used the mean squared error as the loss function optimize.

4.3.2. Modified Hungarian Algorithm

Once we have learned a function to quantify the similarity between detections, we used the similarity values provided by the network to merge repeated detections from the same signs. We accomplished this with a modified version of the Hungarian algorithm [32]. The Hungarian algorithm provides a polynomial time solution to compute the minimum cost in a bipartite graph where each edge has a matching cost. In each pair of consecutive frames from our dataset, we constructed a bipartite graph where each node represents a detection from that image, and each edge connecting two nodes has a weight that indicates the assignment cost for marking those two nodes as belonging to the same sign. The assignment cost of each pair of signs is determined by providing them as input to the similarity network and taking the resulting similarity score as previously described. By using the Hungarian algorithm to compute the assignments of nodes that achieves the minimum sum of costs, similar sign detections as measured by the similarity network are most likely to be paired, and detections with greater pairing cost are less likely to be paired with one another.

One limitation of the Hungarian algorithm is that it always pairs as many nodes from the bipartite graph as possible. For example, if one set in the graph has 5 nodes and the other set contains 4 nodes, the 4 pairings that minimize the sum of costs will be selected by the Hungarian algorithm. This behavior is undesirable for our application, since it is possible for multiple signs to disappear from view between frames and for many new signs appear to appear in the second frame, so pairing as many nodes as possible would result in nodes representing detections from different objects being incorrectly paired. We solve this problem with a simple modification to the algorithm. If the similarity score computed between a pair of detections is greater than a cutoff threshold of 0.7, then the detected objects are forcibly split, meaning the detections will be placed in separate tracklets. The final output of the tracker is a set of tracklets in which each tracklet represents a list of detections predicted to belong to the same sign.

4.3.3. Geo-Localized Sign Prediction

The only remaining step in our pipeline is to condense the tracklets into sign predictions. The simplest method is to predict a sign at the GPS coordinates and with the class from the last frame in the tracklet, which we refer to as the frame of interest (FOI) method. A similarly simple approach is to take a weighted average of the predicted GPS coordinates from each detection in the tracklet. Frames in which the camera is closer to the sign have their predicted GPS weighted more heavily. We predict the class as being the mode of the detections in the tracklet. A third approach involves performing triangulation to condense the tracklets into sign predictions, and predicting the sign class as the mode class from the tracklet. Finally, we can use the Markov random field model proposed in [6] to reduce the tracklets we have produced into sign predictions.

5. Results

5.1. Object Detector Performance

While the ultimate objective of our system is to perform object geo-localization, as an intermediary step we first benchmark the performance of our object detection system. We initialized our object detector with weights from a pre-trained model on the COCO dataset [35]. We kept the default optimization parameters provided by RetinaNet [21]

with the exception of increasing the initial learning rate to 1×10^{-4}. We used smooth L1 loss on both the bounding box regression-subnet and the GPS-subnet. The L1 loss for the GPS subnet is computed relative to the correct offset by transforming the annotated GPS coordinates to the local image coordinate system using the transformation outlined in Section 4. We used our custom focal loss function to train the classification subnet. Our models were trained and tested on a workstation with an NVIDIA 1080ti GPU, as well as a computing cluster with NVIDIA Tesla V100 GPUs. We reported the mean average precision mAP evaluated with an intersection over union IoU = 0.5 on the ARTSv2 dataset. To further illustrate the effect of the proposed FL*e* loss function, we show how the average precision score differs between the worst, 50th percentile, and best performing class. Results are shown in Table 2.

Table 2. Average precision scores on the testing portion of the ARTSv2 dataset. The MAP score indicates the mean of all average precision scores evaluated at an IoU threshold equal to 0.5. We further show average precision scores for the class with the minimum average precision score, the 50th percentile AP score, and the maximum AP score.

Loss Function	mAP_{50}	AP_{min}	$AP_{50\%}$	AP_{max}
RetinaNet-50 (FL)	69.9	15.9	70.0	**100**
RetinaNet-50 (FL*e*)	**70.1**	**17.2**	**70.1**	**100**

5.2. Object Detector GPS Prediction

Each detection produced by the detector has a corresponding offset prediction from the GPS-subnet, which can be transformed to a GPS location using the previously established coordinate transformation. To quantify the performance of this component of our system, we computed the mean absolute error between the location predicted by GPS-RetinaNet and the ground-truth location of the corresponding sign. To construct an error metric easily interpretable by humans, we converted the absolute error between GPS locations to meters using the Haversine formula, which provides accurate approximations at close distances. The Haversine formula is denoted as follows where δ is the relative distance, ψ is latitude, λ is longitude, and R is the mean of earth's radius equal to 6371 km:

$$a = \sin^2\left(\frac{\Delta\psi}{2}\right) + \cos\psi_1 \cdot \cos\psi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right),$$
$$\delta = 2R \cdot atan2\left(\sqrt{a}, \sqrt{1-a}\right).$$

The distribution of mean GPS regression errors for each class is displayed in Figure 10.

5.3. Similarity Network

Next, we quantified the performance of the similarity network, which learns to predict a value closer to 0 if the two input detections belong to the same physical sign and a value closer to 1 if the detections are from different signs. Intuitively, the range of values from 0 to 1 can be interpreted as an abstract measure of "distance" between the two detections. Values closer to 0 indicate the signs are less distant and thus have more in common, whereas values closer to 1 indicate the signs are more distant and thus less similar. Since this network is not performing classification, we can instead quantify its performance by measuring the absolute error at different percentiles. In Table 3, each percentage indicates how often the network predicts a value with an absolute error less than or equal to the listed error value. We use 80% of the annotations for training the network, 10% for validation, and the remaining 10% is reserved for testing.

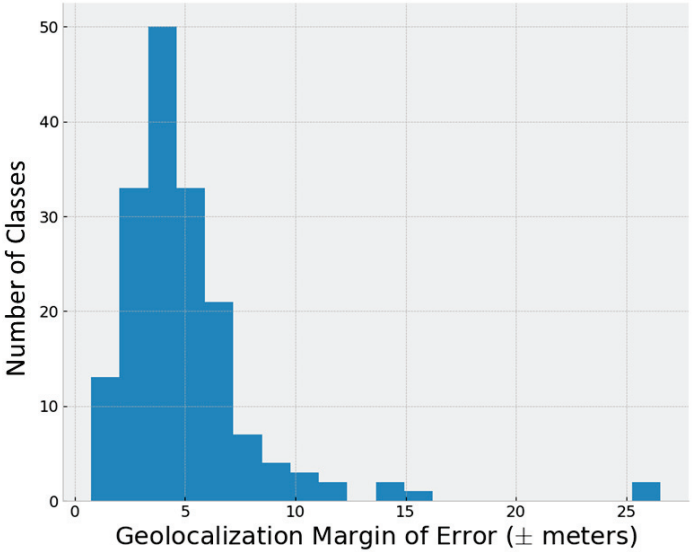


Figure 10. Average GPS testing error for each class. The *x*-axis shows the average geo-localization margin of error of a given class, and the *y*-axis indicates how many classes fell within that approximate margin of error. Our GPS-subnet scored a median MOE of (±5) meters. We can see that the GPS-subnet can accurately estimate distance within a reasonably low margin of error, especially considering how far many signs are from the camera in the ARTSv2 dataset.

Table 3. A table showing the distribution of prediction errors. Each percentile indicates the percent of absolute errors from the similarity network that are at worst equal to the listed error value.

Percentile	Absolute Error
50	0.0165
75	0.1195
90	0.3846
95	0.6106
97	0.7436
98	0.8064
99	0.8844

5.4. Tracker

The objective of the tracker is to collapse down the detections produced by RetinaNet into geo-localized sign predictions. Object geo-localization using deep learning is a new and growing field. There are yet to be any universally accepted performance metrics, especially since performance in this domain is particularly sensitive to the difficulty of the dataset. The goal of our performance evaluation is to quantify how well the physical sign predictions match up with the annotated physical signs distinguished in the ARTS dataset by their integer ID. Specifically, we define a true positive as when the tracker predicts a sign that correctly matches to a real sign within 15 m. We define a false negative as a circumstance where there exists a real sign, but the tracker fails to generate a corresponding prediction. Lastly, we define a false positive to be when the tracker predicts the existence of a sign, but no real-world counterpart exists. An ideal tracker should achieve as many true positives as possible, while minimizing the count of false negatives and false positives.

In Table 4, we show the number of true positives, false negatives, and false positives during different years containing different road segments. The data for geo-localization are divided into years in which they were gathered, and each year contains road segments that the tracker steps through to perform geo-localization. Each individual year is captured in a

variety of geographical regions spread throughout the state of Vermont. Individual years do, however, differ in terms of the driving environments they contain. The 2012 data mostly contain footage from towns, which are challenging due to the density of signs and extra objects present. The 2013 data are composed mostly of highways, and are therefore the least challenging due to having fewer signs, more space between signs, and fewer non-road signs such as signs for businesses. The 2014 data contain many rural segments, which are less challenging, but also contain some towns with difficult sign assemblies.

Figure 11 quantifies the GPS error between each predicted sign and its corresponding ground truth. The *x*-axis shows the GPS error measured in meters, and *y*-axis indicates the probability of a sign being geo-localized within the corresponding mean error.

Table 4. A performance benchmark of the full system end-to-end. Raw images are fed into GPS-RetinaNet, which detects signs, predicts their class, and regresses their GPS offsets. Pairs of detections from consecutive frames predicted by GPS-RetinaNet are fed into the similarity network to predict a similarity score. These similarity scores are provided to the Hungarian algorithm to merge repeated detections of the same sign. These final sign predictions are compared to the annotations in the dataset to determine if they are true positives, false negatives, or false positives. The data are organized into three separate years in which it was gathered. Since each year represents a different set of driving environments, the results are shown separately. The 2012 data contain many towns, 2013 contain mostly highways, and 2014 contain a combination of towns are rural segments. The "All" section shows the combined results for all three years.

End-To-End Performance				
Year Collected	**2012**	**2013**	**2014**	**All**
Noteworthy Features	**Towns**	**Highways**	**Rural Segments and Small Towns**	**All Geographical Environments**
True Positives	264	3170	3179	6163
False Negatives	176	604	842	1622
False Positives	67	826	1581	2474

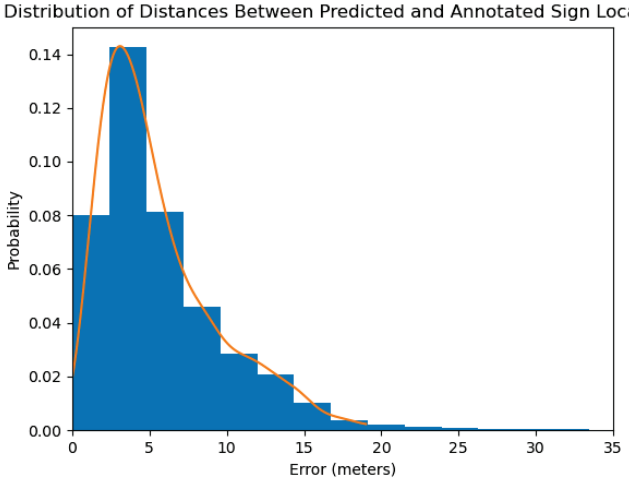


Figure 11. A probability distribution of GPS errors between the predicted geo-localized sign coordinates and the actual coordinates from the annotations. The *x*-axis indicates the amount of GPS error in meters between a predicted and an actual sign, and the *y*-axis indicates the probability of a random sign having the error indicated on the *x*-axis.

5.5. Comparison to Other Methods

Comparison to existing geo-localization techniques is challenging due to the limitations of current approaches, lack of standardized evaluation metrics and varying structure to datasets.

We believe our dataset is the most representative of data encountered by geo-localization systems in the real world; however, this also limits the comparisons we are capable of performing. For example, it is impossible for us to compare our results to [1], since their method uses 5D pose data, which are unavailable in our dataset. Many other tracking methods do not transfer well to our problem either due to not being designed to deal with the very low frame rate or the broad and sparse class distribution contained by the dataset. Other systems also do not take object class into account, and thus are unable to generate complete predictions on our dataset.

While there are not directly analogous state-of-the-art approaches to compare to, we can compare the geo-localization performance of different techniques on our tracklets. Each algorithm receives as input each sequence of detections created by the tracker, and we will compare how effectively the GPS coordinates of each sign can be determined from each of these tracklets. The simplest method is to predict a sign at the GPS coordinates and the sign's class using the last detection in the tracklet, which we refer to as the frame of interest method. The intuition behind this approach is the detection should contain the most accurate class and GPS predictions during the last frame in which the camera is closest to the sign. A similarly simple approach is to take a weighted average of the GPS coordinates from the tracklet, in which images where the camera is closer to the sign are weighted more heavily. The class is predicted to be the mode of the detections in the tracklet. Our third approach is to perform triangularization to condense the tracklets into sign predictions. Finally, we use the Markov random field model proposed in [6] to reduce the tracklets into sign predictions. The results are displayed in Table 5.

Table 5. Performance comparison using different methods to reduce tracklets into sign detections. For each method, we count the total true positives, false negatives, and false postives compared to the ground truth for the full dataset. The mean GPS error indicates the mean absolute distance between a true positive sign prediction and its corresponding ground truth in meters. The STD GPS error indicates the standard deviation of the distribution of true positive GPS errors.

Geo-Localization Performance Comparisons					
Tracking Method	**True Positives**	**False Negatives**	**False Positives**	**Mean GPS Error**	**STD GPS Error**
Triangularization	6079	3000	1918	6.67	4.33
MRF	6677	4379	2156	6.57	4.98
Frame of Interest	6677	2759	1558	5.85	4.40
Weighted Average	6670	2751	1565	5.81	4.38

6. Discussion

6.1. Object Detector Performance

We can see in Table 2 that our proposed FL*e* loss function slightly improves the average mean average precision score of the object detector. The particular difference between these loss functions, however, is that FL*e* demonstrated improved tail performance with greater AP scores for more challenging classes. This result supports the effectiveness of FL*e* in emphasizing low performing classes and ensuring that training gives more weight towards improving their AP. Moreover, FL*e* does not appear to have significantly decreased the mAP or the AP of classes that performed well with FL. This suggests that FL*e* is a sound compromise between promoting poorly performing classes and retaining the performance of easier classes.

6.2. Object Detector GPS Performance

In Figure 10, we show the distribution of the mean prediction errors for each sign class. We observe that most classes have mean predicted distances within 5 m of their labeled ground truth coordinates; however, we note that it is possible the ground truth coordinates themselves could have additional error due to hardware limitations associated with GPS. We observe that the distribution has a right skew due to a few outlier classes with much larger errors. This is largely a consequence of these signs appearing with low frequency in the dataset. Inspection of these difficult classes revealed they corresponded to signs that have a particularly broad distribution in their size, which is unsurprisingly challenging on a data set composed of images from a single camera.

6.3. Similarity Network

As we can see from Table 3, the similarity network achieved a 90th percentile error of approximately 0.38. This means that 90% of predictions it made had an absolute error less than or equal to this value. A total of 95% of the prediction errors were less than 0.61. We can use these values as feedback to decide how we should set our cutoff value for the modified Hungarian algorithm we used. Since we only want to use our cutoff to forcibly split detections when the network is confident they are the not the same sign, this result justifies our decision to use 0.7 as the cutoff threshold.

Visual inspection of failed predictions from the similarity network showed it struggles most with signs that are far away from the camera or similar in appearance to each other. Both these failure cases make intuitive sense because further away and more similar signs will both have fewer visible distinguishing features. Another common failure case for the similarity network is when signs disappear between frames due to being occluded by an object, or are are only partially visible due to being on the edge of the camera's field of view.

6.4. Tracker

Table 4 shows the final performance results of the full end-to-end system broken down by the different years the images from the dataset are organized into. The performance is strongest for 2013. The images from 2013 are captured from the highway, meaning signs tend to be spread further apart. This means the tracker makes fewer errors in combining detections into tracklets, which results in fewer false positives. By contrast, the 2014 data were captured in a combination of rural environments and towns, and therefore have many sign assemblies containing clusters of similar in appearance signs. This additional challenge resulted in greater false positives due to the previously discussed challenge with differentiating between similar signs within clusters.

Manual inspection of false negatives showed that they typically belonged to small signs that are far away or rotated such that they are not directly facing the camera. Due to their lower visibility, it is unsurprising these characteristics increase the likelihood of a sign being undetected. Inspection of false positives shows many of them are caused by detections of other signs that are not actual road signs. For example, a sign from a restaurant may be detected and predicted as a sign, but since this is not technically a traffic sign it is considered a false positive during evaluation. False positives are also caused by other objects with sign-like appearances such as license plates. Finally, inspection of some false positives revealed they were correct detections of actual traffic signs, however they were not annotated as part of the dataset due to being far off in the background of the image or only partially visible in the frame.

6.5. Comparison to Other Methods

We compared the different methods for condensing tracklets into the final sign prediction as is shown in Table 5. The weighted average approach is the most effective method of converting the tracklets into sign predictions. It achieved the lowest GPS error, low standard deviation, and good scores for true positives, false negatives, and false positives. Using the "Frame of Interest" from each tracklet to create the final sign prediction achieved

similar performance. Triangulation has a low standard deviation in its error and is therefore more consistent; however, both triangulation and the MRF approach have greater mean GPS error.

7. Conclusions

In this paper, we presented an enhanced version of the ARTS dataset [9], ARTSv2, which will serve as a comprehensive geo-localization dataset to support future research in the field. Each sign annotation in ARTSv2 consists of a sign class, a side of road indicator, a sign assembly indicator, and a unique sign integer identifier.

We also proposed a novel two-stage object geo-localization system that handles a objects from a large number of heavily skewed classes which exist in an arbitrary number of frames using only accessible hardware. In the first stage, we constructed an object detector called GPS-RetinaNet, which predicts bounding box coordinates, sign classes, and GPS offsets for each detected sign in an input image. GPS-RetinaNet uses FLe, a novel variant of focal loss, during training to effectively handle the class imbalance present in ARTSv2.

The second stage of our proposed modes is a novel object tracking system to collapse a set of detections in a noisy, low-frame rate environment into final geo-localized object predictions. The traffic sign tracking and geo-localization was handled using a learned metric network and a variant of the Hungarian algorithm.

Future research should explore optimizations and tuning to facilitate high frame rate object geo-localization.

The noise introduced to GPS coordinates due to both equipment error and annotation inconsistencies limits the capability of GPS to serve as a ground truth. To limit GPS error, future work could use satellite images to achieve enhanced geo-localization performance. Future work could also experiment with how to better distinguish between signs with similar visual features and locations during tracking, as these objects have the fewest distinguishing features.

Author Contributions: Conceptualization, D.W., T.A., C.V.O., X.Z., S.W. and J.N.; methodology, D.W., T.A. and C.V.O.; software, D.W., T.A. and C.V.O.; validation, D.W., T.A. and C.V.O.; formal analysis, D.W., T.A. and C.V.O.; investigation, D.W., T.A., C.V.O. and S.W.; resources, S.W. and J.N.; data curation, D.W., S.W. and J.N.; writing—original draft preparation, D.W., T.A., C.V.O., X.Z. and S.W.; writing—review and editing, D.W. and S.W.; visualization, D.W. and T.A.; supervision, S.W. and J.N.; project administration, S.W. and J.N.; funding acquisition, S.W. and J.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Vermont Agency of Transportation.

Data Availability Statement: We have made the ARTSv2 dataset publicly available to support research and development in both traffic sign recognition and object geo-localization. ARTSv2 can be accessed at the following https://drive.google.com/drive/u/1/folders/1u_nx38M0_owB0cR-qA6IOWgZhGpb9sWU (accessed on 5 April 2022). Source code is also available, and can be accessed at the following https://gitlab.com/vail-uvm/VTrans-AI (accessed on 5 April 2022).

Acknowledgments: Computations were performed using the Vermont Advanced Computing Core supported in part by NSF award No. OAC-1827314. We thank Josh Minot and Fayha Almutairy for their contributions, discussions, and feedback on this project. This work would not have been possible without great collaboration between the Vermont Artificial Intelligence Lab and the Vermont Transportation Agency. We would like to thank Rick Scott and Ken Valentine for championing this project.

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

Appendix A

Focal Loss was first introduced in [21] to address the challenge of overwhelming the loss value of rare classes with many easy classes during training for datasets with unequally distributed samples. One of the most crucial properties of the FL is the basic

idea of down-weighting the loss of easy (well-classified) classes in favor of focusing the training on the hard classes in the dataset. Focal Loss is defined as:

$$\mathrm{FL}(p_t) = -(1-p_t)^{\gamma}\log(p_t). \tag{A1}$$

The focusing parameter γ acts as a modulator to fine-tune the effect of down-weighting the loss of easy classes. Ref. [21] noted that $\gamma = 2$ works well, since it maintains acceptable performance on easy classes while noticeably improving performance on hard classes. In our experiments, however, we found that fixing the focusing parameter value for all classes results in an unintended effect in which the loss value of a wide range of classes starts to become down-weighted prematurely, not allowing them to achieve better average precision in a reasonable amount of time. In other words, FL increasingly down-weights the loss value of all classes once their probability p_t surpasses 0.3, which one can argue that it is too low to consider as a threshold for 'well-classified' classes.

We propose a modification to Focal Loss that replaces γ in the original definition by an adaptive modulator. We define the new focusing parameter as:

$$\Gamma = e^{(1-p_t)}, \tag{A2}$$

$$\mathrm{FL}e(p_t) = -(1-p_t)^{\Gamma}\log(p_t). \tag{A3}$$

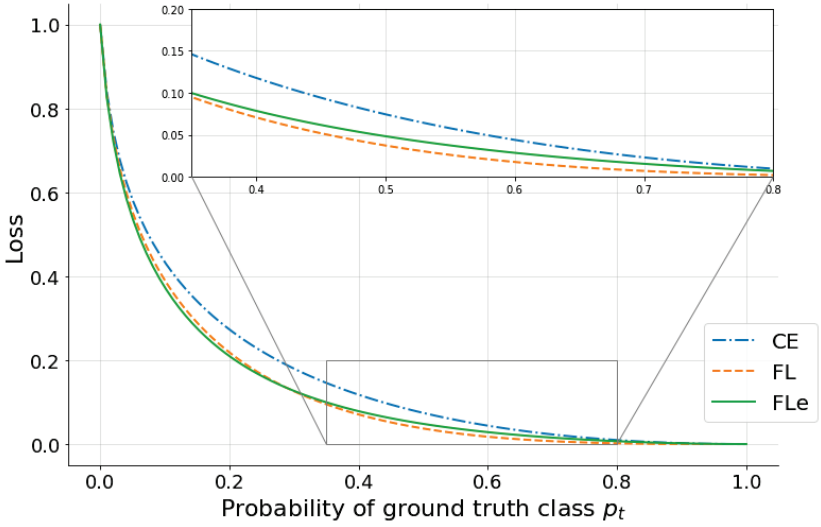


Figure A1. Our modified Focal Loss function (FL*e*) compared with FL ($\gamma = 2$), and cross entropy (CE). FL*e* introduces an adaptive exponent to the original FL [21]. This effectively changes the underlying distribution of classes in regards to their APs and promotes some of the poorly classified classes to a better score while preserving the performance of well-classified classes.

For convenience, we will refer to our new definition of Focal Loss as (FL*e*). FL*e* introduces two new properties to the original definition. It dynamically fine-tunes the exponent based on the given class performance to reduce the relative loss for well-classified classes maintaining the primary benefit of the original FL. Figure A1 directly compares FL with FL*e*, highlighting that FL*e* (shown in green) crosses over $FL_{\gamma=2}$ (shown in orange) around ($p_t = 0.3$). As p_t goes up from $0.3 \rightarrow 1$, FL*e* starts to shift up slowly ranging in between FL and Cross Entropy CE (shown in blue).

In practice, this allows us to ultimately define 'well-classified' classes as ($p_t > 0.7$) instead of ($p_t > 0.3$) in the original definition. In other words, FL*e* reduces the loss down-weighting effect on classes when their p_t values are in the range ($0.3 \geq p_t \geq 0.7$) while still

focusing on hard classes. This results in slightly improved performance that manifests at the beginning of the training and continues throughout the process until both FL and FL*e* converges at a similar mAP; however, FL*e* will have a slightly lower standard deviation as more classes will cluster around mAP whereas FL will have a greater spread of APs per class.

Appendix B

We argued in Section 2 that traditional trackers were ineffective in the object geo-localization domain due to not being designed for low frame rate datasets and not taking GPS information into consideration during tracking. In Table A1, we tested several popular object trackers and verified that they provide extremely poor results. As stated, they are unable to track objects due to how far apart frames are, leading them to nearly always predict two objects as being "different." Since objects are rarely predicted to be the same by traditional trackers, repeated occurrences of objects are not merged, leading to extremely high false positive rates.

Table A1. Performance using different trackers to condense repeated detections from GPS-RetinaNet. Other methods essentially fail completely to merge repeated detections, since they nearly always predict detections from separate frames are different signs. This occurs because they are not designed to handle large "jumps" in object's positions and angles between frames.

Tracker Performance Comparisons			
Tracker	**True Positives**	**False Negatives**	**False Positives**
Boosting [36]	8062	173	24,425
MIL [37]	8068	167	24,130
KCF [38]	8061	175	25,812
TLD [39]	8055	180	21,903
MedianFlow [40]	8054	181	20,834
GoTurn [41]	8049	186	22,203
MOSSE [42]	8042	193	21,422
CSRT [43]	8061	174	23,052
Proposed Tracker	**6677**	**2759**	**1558**

References

1. Chaabane, M.; Gueguen, L.; Trabelsi, A.; Beveridge, R.; O'Hara, S. End-to-End Learning Improves Static Object Geo-Localization From Video. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Virtual, 5–9 January 2021; pp. 2063–2072.
2. Nassar, A.S.; Lefèvre, S.; Wegner, J.D. Simultaneous multi-view instance detection with learned geometric soft-constraints. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6559–6568.
3. Nassar, A.S.; D'Aronco, S.; Lefèvre, S.; Wegner, J.D. GeoGraph: Graph-Based Multi-view Object Detection with Geometric Cues End-to-End. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 488–504.
4. McManus, C.; Churchill, W.; Maddern, W.; Stewart, A.D.; Newman, P. Shady dealings: Robust, long-term visual localisation using illumination invariance. In Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 901–906. [CrossRef]
5. Suenderhauf, N.; Shirazi, S.; Jacobson, A.; Dayoub, F.; Pepperell, E.; Upcroft, B.; Milford, M. Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free. In Proceedings of the Robotics: Science and Systems XI, Rome, Italy, 13–17 July 2015; pp. 1–10.
6. Krylov, V.A.; Kenny, E.; Dahyot, R. Automatic Discovery and Geotagging of Objects from Street View Imagery. *Remote Sens.* **2018**, *10*, 661. [CrossRef]
7. Krylov, V.A.; Dahyot, R. Object geolocation using mrf based multi-sensor fusion. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 2745–2749.
8. Wilson, D.; Zhang, X.; Sultani, W.; Wshah, S. Visual and Object Geo-localization: A Comprehensive Survey. *arXiv* **2021**, arXiv:2112:15202.
9. Almutairy, F.; Alshaabi, T.; Nelson, J.; Wshah, S. ARTS: Automotive Repository of Traffic Signs for the United States. *IEEE Trans. Intell. Transp. Syst.* **2019**, *22*, 457–465. [CrossRef]

10. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117. [CrossRef]
11. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2010; pp. 307–312.
12. Fairfield, N.; Urmson, C. Traffic light mapping and detection. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 5421–5426.
13. Soheilian, B.; Paparoditis, N.; Vallet, B. Detection and 3D reconstruction of traffic signs from multiple view color images. *ISPRS J. Photogramm. Remote Sens.* **2013**, *77*, 1–20. [CrossRef]
14. Hebbalaguppe, R.; Garg, G.; Hassan, E.; Ghosh, H.; Verma, A. Telecom Inventory management via object recognition and localisation on Google Street View Images. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 725–733.
15. 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 (CVPR'05), San Diego, CA, USA, 21–23 September 2005; Volume 1, pp. 886–893.
16. Liu, C.J.; Ulicny, M.; Manzke, M.; Dahyot, R. Context Aware Object Geotagging. *arXiv* **2021**, arXiv:2108.06302.
17. Lin, T.; 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]
18. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
19. Girshick, R. Fast R-CNN Object detection with Caffe. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chilie, 7–13 December 2015; pp. 1440–1448. [CrossRef]
20. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
21. Lin, T.; Goyal, P.; Girshick, R.B.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2018**, arXiv:1708.02002.
22. He, K.; Gkioxari, G.; Dollár, 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.
23. Zhu, J.; Yang, H.; Liu, N.; Kim, M.; Zhang, W.; Yang, M.H. Online multi-object tracking with dual matching attention networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 366–382.
24. Voigtlaender, P.; Krause, M.; Osep, A.; Luiten, J.; Sekar, B.B.G.; Geiger, A.; Leibe, B. Mots: Multi-object tracking and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7942–7951.
25. Son, J.; Baek, M.; Cho, M.; Han, B. Multi-object tracking with quadruplet convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5620–5629.
26. Xu, J.; Cao, Y.; Zhang, Z.; Hu, H. Spatial-temporal relation networks for multi-object tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 3988–3998.
27. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the Computer Vision—ECCV 2016 Workshops, Amsterdam, The Netherlands, 11–14 October 2016; Hua, G., Jégou, H., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 850–865.
28. Xiang, Y.; Alahi, A.; Savarese, S. Learning to Track: Online Multi-object Tracking by Decision Making. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4705–4713. [CrossRef]
29. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11621–11631.
30. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
31. Tzutalin. Tzutalin. LabelImg. Git Code. 2015. Available online: https://github.com/tzutalin/labelImg (accessed on 5 April 2022).
32. Kuhn, H.W. The Hungarian Method For The Assignment Problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. doi: 10.1002/nav.3800020109. [CrossRef]
33. 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. [CrossRef]
34. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980.
35. Lin, T.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. *arXiv* **2014**, arXiv:1405.0312.
36. Grabner, H.; Grabner, M.; Bischof, H. Real-Time Tracking via On-line Boosting. In Proceedings of the British Machine Vision Conference 2006, Edinburgh, UK, 4–7 September 2006; Volume 1, pp. 47–56. [CrossRef]

37. Babenko, B.; Yang, M.H.; Belongie, S. Visual tracking with online Multiple Instance Learning. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 983–990. [CrossRef]
38. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef]
39. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-Learning-Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1409–1422. [CrossRef]
40. Kalal, Z.; Mikolajczyk, K.; Matas, J. Forward-Backward Error: Automatic Detection of Tracking Failures. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2756–2759. [CrossRef]
41. Held, D.; Thrun, S.; Savarese, S. Learning to Track at 100 FPS with Deep Regression Networks. *arXiv* **2016**, arXiv:1604.01802.
42. Bolme, D.; Beveridge, J.; Draper, B.; Lui, Y. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550. [CrossRef]
43. Lukežič, A.; Vojíř, T.; Čehovin Zajc, L.; Matas, J.; Kristan, M. Discriminative Correlation Filter with Channel and Spatial Reliability. *Int. J. Comput. Vis.* **2018**, *126*; 671–688. [CrossRef]

remote sensing

MDPI

Technical Note

An Optimal Transport Based Global Similarity Index for Remote Sensing Products Comparison

Yumin Tan , Yanzhe Shi *, Le Xu, Kailei Zhou, Guifei Jing, Xiaolu Wang and Bingxin Bai

School of Transportation Science and Engineering, Beihang University, No. 37 Xueyuan Road, Haidian District, Beijing 100191, China; tanym@buaa.edu.cn (Y.T.); xulele@buaa.edu.cn (L.X.); zhoukailei@buaa.edu.cn (K.Z.); guifeijing@buaa.edu.cn (G.J.); zy2113207@buaa.edu.cn (X.W.); baibx@buaa.edu.cn (B.B.)
* Correspondence: syz_cannot@buaa.edu.cn

Abstract: Remote sensing products, such as land cover data products, are essential for a wide range of scientific studies and applications, and their quality evaluation and relative comparison have become a major issue that needs to be studied. Traditional methods, such as error matrices, are not effective in describing spatial distribution because they are based on a pixel-by-pixel comparison. In this paper, the relative quality comparison of two remote sensing products is turned into the difference measurement between the spatial distribution of pixels by proposing a max-sliced Wasserstein distance-based similarity index. According to optimal transport theory, the mathematical expression of the proposed similarity index is firstly clarified, and then its rationality is illustrated, and finally, experiments on three open land cover products (GLCFCS30, FROMGLC, CNLUCC) are conducted. Results show that based on this proposed similarity index-based relative quality comparison method, the spatial difference, including geometric shapes and spatial locations between two different remote sensing products in raster form, can be quantified. The method is particularly useful in cases where there exists misregistration between datasets, while pixel-based methods will lose their robustness.

Keywords: similarity comparison; Wasserstein distance; raster; land cover

Citation: Tan, Y.; Shi, Y.; Xu, L.; Zhou, K.; Jing, G.; Wang, X.; Bai, B. An Optimal Transport Based Global Similarity Index for Remote Sensing Products Comparison. *Remote Sens.* **2022**, *14*, 2546. https://doi.org/10.3390/rs14112546

Academic Editors: Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Received: 6 May 2022
Accepted: 24 May 2022
Published: 26 May 2022

1. Introduction

With the quick and great development of remote sensing technologies, together with the spread of open science, there are many available similar remote sensing products, and global land cover products are very typical examples, which are typically presented as digital thematic maps in raster. Thus, to facilitate common users' easy-to-choose appropriate products, the need to compare their accuracy is growing. An ideal accuracy assessment is based on comparing a dataset with its true value; however, usually, it is impossible to obtain 'ground truth' in practice, thus accuracy assessments are usually conducted by comparing the dataset with some 'reference data'. We know that remote sensing products from different sources are different in many aspects, including data source, classification scheme, methodology and resolution, etc. [1]. This is not surprising, given the fact that quantitative analyses of complex land cover types remain an arduous task [2].

2. Related Works

Table 1 lists some commonly used relative comparison methods of remote sensing products. Considering their definition, processing units and evaluating indicators, relative comparison methods can be classified into three categories: error matrix-based, local spatial feature-based and others. In these methods, spatial features are expressed at three scales: (i) a local scale for statistical analysis of pixels, (ii) a global scale for analysis of the whole image, and (iii) specific scopes, such as sliding windows.

Table 1. Relative comparison methods.

Comparison Methods		Processing Unit	Qualitative/Quantitative	Evaluating Indicator	Attention Scale
Error Matrix-based methods	Pixel-by-pixel based Statistical method [3–5]	pixel	qualitative	OA, UA, PA, kappa coefficient, information entropy, etc.	Local scale
			quantitative	Mean, standard deviation, entropy, correlation coefficient, Tau coefficient, etc.	
	Quantity and location-based method [6]		qualitative	Location-based kappa coefficient, quantity-based kappa coefficient, etc.	
local spatial feature-based methods	Spatial distribution-based method [7–9]	category	qualitative	Goodman–Kruskal Cramér's V statistics Theil's U statistics	Global scale
	Neighborhood-based comparison method [10]			Spatial structure and overlap index	
Other methods	Fuzzy comparison [11,12]	pixel and category	qualitative	Fuzzy Kappa coefficient fuzzy similarity index.	Specific scope
	Curvature-fit based method [13,14]	category		Polygon matching index	Specific scope
	Sliding-window based method [15]	sliding window	quantitative	Euclidean distance, correlation coefficient	

Existing relative comparison methods are mostly based on the confusion or error matrix method [16,17]. However, error matrix-based methods ignore the underlying geometry of the space. For example, the blue pixels in Figure 1 represent water bodies. It is clear that the error matrix, user's accuracy, and producer's accuracy in datasets 2 and 3 are the same (because the number of water pixels is the same). However, it is obvious that the spatial positions of the two different areas in datasets 2 and 3 are different compared to dataset 1, and the difference between dataset 2 and dataset 1 is more like a real water body than an error. Therefore, the spatial distribution of "errors" and the information contained in the "errors" are also very important in a relative quality evaluation system. Moreover, the validation techniques based on pixel statistics rely heavily on probability sampling design for collecting validation data [3]. In the error matrix-based methods, reference data are taken as real data, and some studies have shown that if errors in the reference data are related to the predicted data, the comparison accuracy will be overestimated, and if the error is conditional independent, the accuracy will be underestimated [18]. The kappa coefficient based on the error matrix is also considered to be unsuitable [19].

Figure 1. Three datasets of water body distribution (areas with differences are circled in red).

Researchers have proposed some improved methods based on the error matrix. For example, Enøe et al. [20] used a maximum likelihood method to deal with unknown binary reference data. However, this method can only verify the reference data and cannot revise the result. The authors of [21] proposed to simulate the geometric deviation in different directions with a certain step size based on the reference image, and then calculate the geometric accuracy of coarse-resolution remote sensing data by calculating the correlation between the migrated reference image and the image to be evaluated, while there are still uncertainties in determining the size of step. To sum up, these methods based on error matrix lack the description of the spatial structure of remote sensing products.

In order to comprehensively consider and quantify the possible spatial features in different remote sensing products in the form of raster datasets, we attempted to design a "similarity index" by taking raster datasets as the probability distributions in a two-dimensional space, and then measured the difference between two distributions, thus the problem of accuracy comparison between different raster datasets is turned into a multi-category optimal transmission question. Therefore, in fact, it is now a question of spatial similarity measurement.

Optimal transport theory gives a good framework for comparing two measures in a Lagrangian framework, and Wasserstein distance is an important concept arising from optimal transport, which are the metrics of probability distributions. At present, some applications of optimal transport theory in the field of remote sensing have been proposed [22] to fuse remote sensing products with social media information by using the natural interpretation of distribution distance in Wasserstein metric space [23] using high-resolution satellite time-series images to evaluate the accuracy of remote sensing mapping products in the absence of field verification data by using EMD transmission and Sinkhorn transmission.

The optimal transport theory establishes a geometric tool for effectively comparing probability distributions. The relative similarity concept in the paper is based on max-sliced Wasserstein distance [24]. The main contributions of this paper are the following:

(i) Category information contained in multi-source raster datasets is treated as a probability distribution of spatial information in a 2D space, and then the problem of consistency measurement between remote sensing products is converted into a measurement question of probability distribution.
(ii) A max-sliced Wasserstein distance-based similarity index is designed and calculated, which could solve the product comparison problem in the case of misregistration.

3. Methodology

Wasserstein distance provides a way to measure the distance between two non-empty datasets, and a raster dataset can be taken as a set of multiple types of pixel coordinates, so this measurement applies.

Wasserstein-P distance between two pixel-based distributions q_1 and q_2 could be expressed as follows:

$$W_P(q_1, q_2) = \left(\inf_{\gamma(x,y) \in \Gamma(q_1, q_2)} E_{(x,y) \sim \gamma(x,y)} \left[d\left(x, y\right)^P \right] \right)^{\frac{1}{P}} \quad (1)$$

where x,y are the distribution of pixel coordinates in the two raster datasets, respectively, $\Gamma(q_1, q_2)$ is the set of all possible joint distributions on (x,y) with marginals q_1 and q_2, d(x,y) is the distance metric between x and y, generally, the Euclidean distance is taken. In a two-dimensional case, $P = 2$.

Figure 2a illustrates the one-dimensional case, that is, using the minimum cost to convert one distribution into another distribution. Figure 2b is a two-dimensional case, using a transmission matrix P to describe the transmission plan.

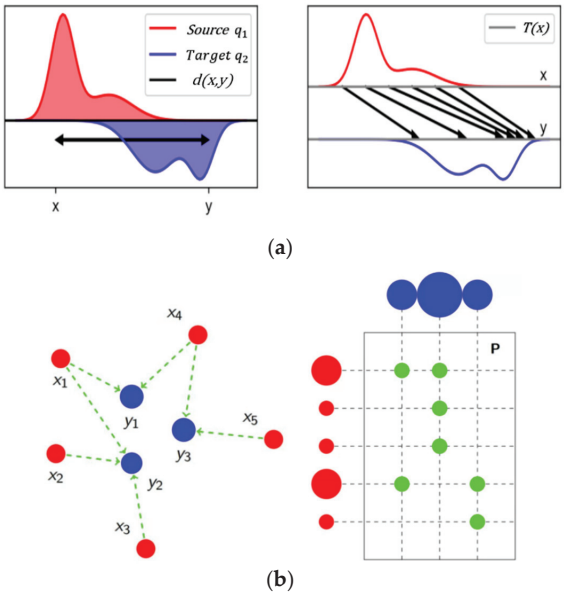


Figure 2. Optimal transport problems: (**a**) one-dimensional case; (**b**) two-dimensional case.

In fact, it is difficult to calculate the Wasserstein distance in two or more dimensions directly through the optimal transport theory. Kantorovich–Rubinstein duality can be used to calculate its one-dimensional case:

$$W(q_1, q_2) = \sup_{\|f\|_L \leqq 1} E_{x\sim q_1}[f(x)] - E_{x\sim q_2}[f(x)] \tag{2}$$

where the supremum is over all the 1-Lipschitz functions $f\ \mathrm{X} \rightarrow \mathrm{R}$. The function f is commonly represented via a deep net and various ways have been suggested to enforce the Lipschitz constraint [25].

Then, a sliced version of the Wasserstein-2 distance, proposed by [26], shows its advantage, which only requires estimating distances of one-dimensional distributions and is more efficient. The "sliced Wasserstein-p distance" between distributions q_1, q_2 is defined as:

$$\widetilde{W}_{\mathrm{p}}(q_1, q_2) = \left[\int_{\omega\in\Omega} W_P^p(q_1^\omega, q_2^\omega) d\omega \right]^{\frac{1}{p}} \tag{3}$$

where q_1^ω, q_2^ω denote the projection (i.e., marginal) of q_1, q_2 onto the direction ω, and Ω is the set of all possible directions on the unit sphere.

The $\widetilde{W}_{\mathrm{p}}(q_1, q_2)$ distance has important practical implications: provided that the projected distributions q_1^ω, q_2^ω can be computed, then for any $\omega \in \Omega$, the distance $\widetilde{W}_{\mathrm{p}}(q_1, q_2)$, as well as its optimal transport map and the corresponding Kantorovich potential can be analytically computed by using projected measures that are one-dimensional.

For two given datasets $\boldsymbol{D} = \{(\mathrm{x})\}$ of samples $\mathrm{x} \sim q_1$, $\boldsymbol{F} = \{(\mathrm{y})\}$ of samples $\mathrm{y} \sim q_2$:

$$W_2^2(\boldsymbol{D}^\omega, \boldsymbol{F}^\omega) = \frac{1}{|\boldsymbol{D}|} \sum_{i=1}^{|\boldsymbol{D}|} \|\boldsymbol{D}^\omega_{\varphi\boldsymbol{D}(i)} - \boldsymbol{F}^\omega_{\varphi\boldsymbol{F}(i)}\|_2^2 \tag{4}$$

where $\varphi\boldsymbol{D}$ and $\varphi\boldsymbol{F}$ are permutations that sort the projected sample sets $\boldsymbol{D}^\omega$ and $\boldsymbol{F}^\omega$, respectively.

$$\boldsymbol{D}^\omega_{\varphi\boldsymbol{D}(1)} \leq \boldsymbol{D}^\omega_{\varphi\boldsymbol{D}(2)} \leq \cdots \leq \boldsymbol{D}^\omega_{\varphi\boldsymbol{D}(|\boldsymbol{D}|)} \tag{5}$$

$$\boldsymbol{F}^{\omega}_{\varphi \boldsymbol{F}(1)} \leq \boldsymbol{F}^{\omega}_{\varphi \boldsymbol{F}(2)} \leq \cdots \leq \boldsymbol{F}^{\omega}_{\varphi \boldsymbol{F}(|\boldsymbol{F}|)} \tag{6}$$

When the number of elements in the two datasets $|\boldsymbol{D}|$,$|\boldsymbol{F}|$ are different, find the greatest common divisor θ of $|\boldsymbol{D}|$ and $|\boldsymbol{F}|$, then make $|\boldsymbol{D}|^* = \frac{|\boldsymbol{F}|}{\theta}$, $|\boldsymbol{F}|^* = \frac{|\boldsymbol{D}|}{\theta}$, and replace $\boldsymbol{D}(i)$ with $|\boldsymbol{D}|^*$ elements, together with replacing $\boldsymbol{F}(i)$ with $|\boldsymbol{F}|^*$ elements. Because $|\boldsymbol{D}|^*|\boldsymbol{D}| = |\boldsymbol{F}|^*|\boldsymbol{F}|$, it is sure that the element numbers in the two new sets are the same.

If we could find the most meaningful projection direction, the 2D max-sliced-Wasserstein distance will be calculated, and it is defined as follows:

$$max - \widetilde{W}_2(q_1, q_2) = \left[max_{\omega \in \Omega} W_2^2(q_1^{\omega}, q_2^{\omega}) \right]^{\frac{1}{2}} \tag{7}$$

This metric satisfies the properties of non-negativity, the identity of indiscernible, symmetry, and subadditivity [27]. Hence, it is a true metric.

Taking two land cover datasets, A and B, with K categories and the size of Row × Col as an example, after each $max - \widetilde{W}_2$ is calculated by categories, the similarity between A and B can be defined:

$$Similarity(A, B) = \sum_{i=1}^{K} [1 - \frac{1}{\sqrt[2]{\mathrm{Row}^2 + \mathrm{Col}^2}} \times min(\sqrt[2]{\mathrm{Row}^2 + \mathrm{Col}^2}, \frac{max - \widetilde{W}_2(i)}{1 - \frac{N(i)}{\mathrm{Row} \times \mathrm{Col}}})] \times \frac{N(i)}{\mathrm{Row} \times \mathrm{Col}} \tag{8}$$

where $max - \widetilde{W_2}(i)$ is the max-sliced Wasserstein distance of category i, and $N(i)$ is the average number of pixels of category i in datasets A and B, that is, $N(i) = \frac{1}{2}[N_A(i) + N_B(i)]$.

It could also be deduced that the similarity between two datasets is order independent, so we have:

$$Similarity(A, B) = Similarity(B, A) \tag{9}$$

This method has the following advantages: (1) It can quantify the difference in both spatial position and shape, then measure it under a unified standard; (2) it can give a continuous transformation process while preserving geometric features; (3) it has symmetry and can still give reasonable measurement results in the case of regional misregistration.

4. Experiment

To demonstrate the rationality of the above-proposed similarity index, we designed several validation experiments both on test datasets and real datasets. For test datasets, we set four cases, and use a gradient descent method to give the continuous transformation process. For real datasets, we calculate the similarity index among three open land cover products.

4.1. Experiments on Test Datasets

(i) Max-sliced Wasserstein distance between two points

Suppose there are two points on the two-dimensional plane, the coordinates of the two points are (9.0, 9.0) and (13.0, 12.0), respectively.

The results of the max-sliced Wasserstein distance with the number of projections and the percentage difference with the Euclidean distance are shown in Table 2.

Table 2. Distance with the number of projections.

Number of projections	5	10	15	20
Max-Wasserstein distance	4.9700	4.9961	4.9970	4.9994
Difference	0.6%	0.078%	0.06%	0.012%

The Euclidean distance between the two points is 5.0. Because the sliced distance is calculated by using projection, there is a deviation from the Euclidean distance. We consider that when the number of projections is not less than 20, the deviation is within the allowable range.

(ii) Max-sliced Wasserstein distance between areas

The geometric shape and spatial position difference between point sets (usually represented as polygons in a raster dataset) cannot be simply measured by the Euclidean distance between points. Here, we design four cases to illustrate the rationality of the proposed index:

Case 1: Shapes of the polygons in two datasets to be compared are the same, and the spatial position has a translation transformation;

Case 2: The centroid of polygons is the same, but the shapes are different;

Case 3: The polygon shapes and spatial positions are both different;

Case 4: The distribution is not continuous and there are multiple areas.

Then, we used the gradient descent method to construct the continuous transformation process.

The initial states of the four examples are shown in Table 3 and Figure 3.

Table 3. Initial state of four cases.

Case.	Distribution		Centroid		Radius/Side Length	
	A	B	A	B	A	B
1	Circle	Circle	(160,160)	(220,160)	60	60
2	Circle	Square	(160,160)	(160,160)	60	60
3	Circle	Square	(160,160)	(220,160)	60	60
4	3 Circles	Circle	(70,70),(70,250),(250,70)	(160,160)	50	60

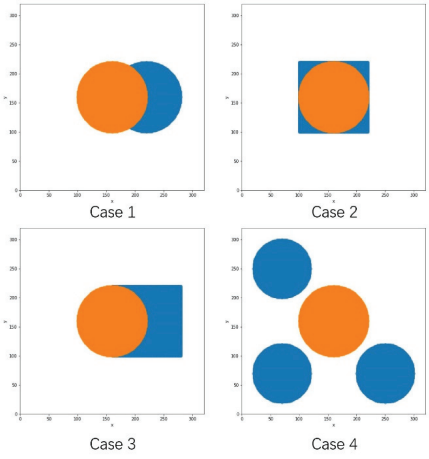


Figure 3. Initial state of four cases.

The projection number is set to 20, and then the max-sliced Wasserstein distances of each case are shown in Table 4.

Table 4. Max-sliced Wasserstein distance of four cases.

Case	1	2	3	4
Distance	59.9964	5.6467	61.2305	77.3263

Distance changes during the gradient descent process are shown in Figure 4. As the number of iterations increases, the distance tends to decrease, which means that distribution is shifted to the target distribution.

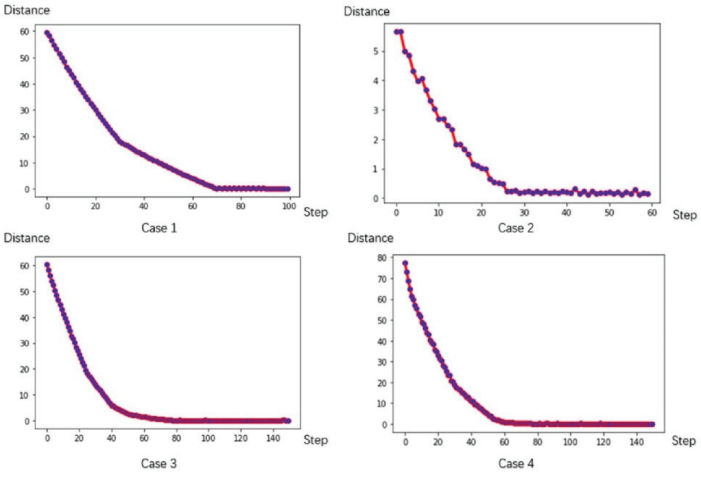


Figure 4. Distance with steps.

The process of continuous transformation of four cases in the gradient descent process is shown in Figure 5.

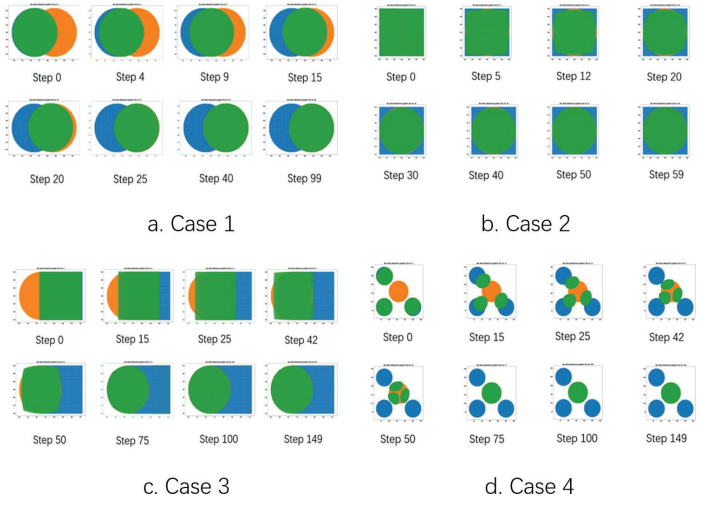


Figure 5. Transformation process of source distribution to target.

In Case 1, the shapes of the two polygons are the same, and the max-sliced Wasserstein distance measures their difference in spatial position. When it goes to the 99th step, the initial distribution is transferred to the target distribution (the green circle moves to the same position as the orange circle).

In Case 2, the positions of the two centroids are the same, so the factor that affects the max-sliced Wasserstein distance measurement is only their geometric shapes.

In Case 3, both their geometric shapes and spatial positions affect the final measurement, so compared with that in Case 1 and Case 2, the distance is the largest.

In Case 4, the distribution of the points is discontinuous, and there are multiple parts (blue points). This simulates a more complicated situation in raster datasets. This metric gives a reasonable result and a continuous transformation process of multiple regions.

4.2. Experiments on Real Remote Sensing Products

We choose open-source global land-use raster datasets to verify the proposed index, and the three selected datasets are GLC_FCS30 [28], FROM_GLC [29], CNLUCC [30] in 2015.

The selected area is a rectangle with the coordinates range of (116.18575, 40.00125) and (116.37300, 40.18375) in the coordinate system of WGS84. The three datasets are the land-use data of this area at the same time (2015), created by different researchers, with the same resolution of 30 m.

4.2.1. Dataset Preprocessing

The classification systems of the three datasets are shown in Figure 6:

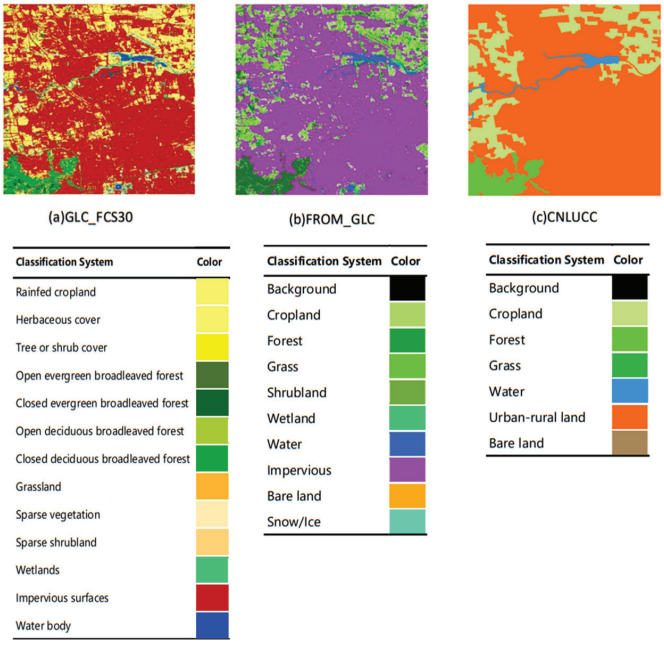
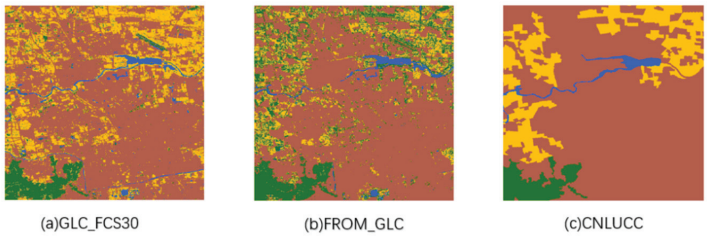


Figure 6. Detail of GLC_FCS30, FROM_GLC and CNLUCC datasets.

Comparisons cannot be made under the original non-uniform classification system, so we reclassified the pixels and classified them into four types: (1) Cropland; (2) Forest/Grass; (3) Water body; (4) Impervious surfaces. Figure 7 illustrates the classification system and reclassification results.



LC id	Classification System		Color
0	Background	(0,0,0)	
1	Cropland	(255,192,0)	
2	Forest/Grass	(38,115,0)	
3	Water	(0,92,255)	
4	Impervious surfaces	(179,94,77)	

Figure 7. Classification System and results of reclassification.

Table 5 shows the pixel numbers of four classification types in each dataset.

Table 5. The number of pixels of four types.

	Cropland	Forest/Grass	Water	Impervious Surfaces	Total
GLC_FCS30	123,957	39,076	12,966	370,771	546,770
FROM_GLC	73,795	92,438	11,936	368,601	546,770
CNLUCC	95,176	26,149	8522	416,923	546,770

4.2.2. Similarity Calculation

Taking water bodies as an example, distributions of water bodies on the three datasets are shown in Figure 8.

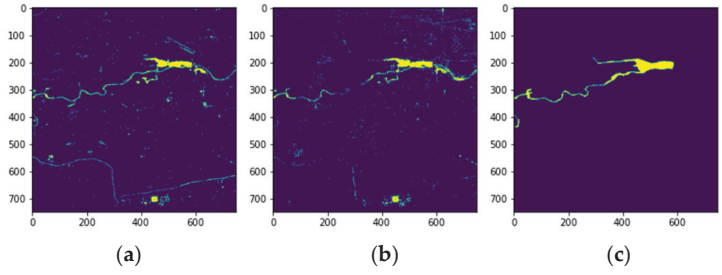


Figure 8. Water body distribution. (**a**) GLC_FCS30; (**b**) FROM_GLC; (**c**) CNLUCC.

Compare the three datasets in pairs separately to calculate the max-sliced Wasserstein distance of the water distribution (Figure 9 and Table 6):

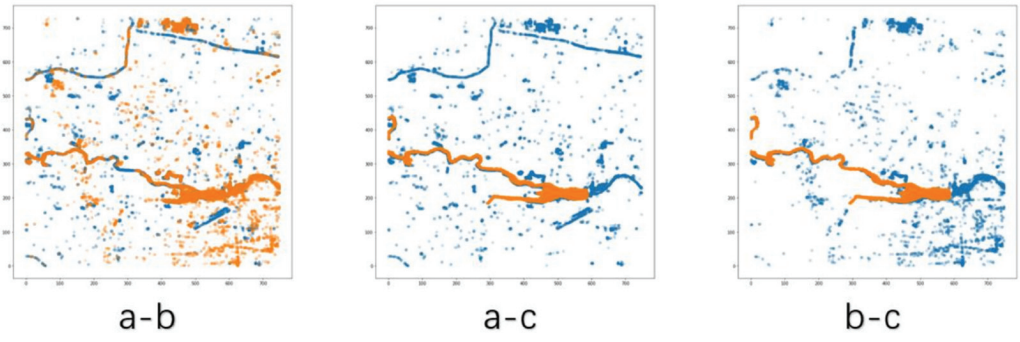


Figure 9. Overlay of raster data map frames (**a-b**. GLC_FCS30-FROM_GLC; **a-c**. GLC_FCS30-CNLUCC; **b-c**. FROM_GLC-CNLUCC).

Table 6. Results of water body.

	Distance	Row × Col	Similarity
a-b	86.9853	749 × 730	91.49%
a-c	180.1471	749 × 730	82.43%
b-c	153.8637	749 × 730	85.01%

The results show that the comparison between GLC_FCS30 and FROM_GLC products (a-b) has the smallest max-sliced Wasserstein distance and thus, the highest similarity, which is consistent with the visual judgment.

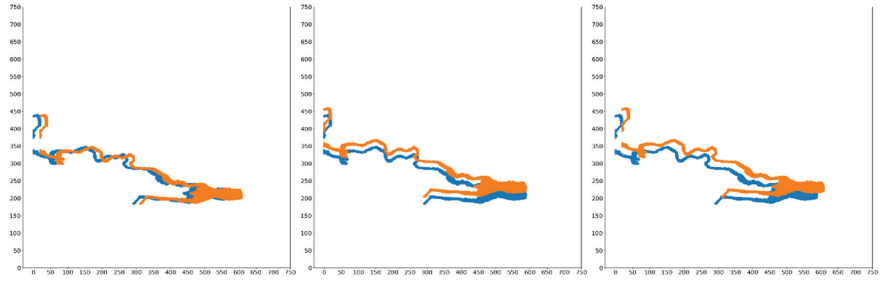
The multi-category comparison results among the three chosen products are listed in Table 7:

Table 7. Results of three datasets.

		a-b	a-c	b-c
Cropland	Distance	26.9367	93.0022	71.9183
	Similarity	96.85%	88.88%	91.87%
Forest/Grass	Distance	232.7238	260.1100	423.1698
	Similarity	74.71%	72.11%	54.62%
Water	Distance	86.9853	180.1471	153.8637
	Similarity	91.49%	82.43%	85.01%
Impervious surfaces	Distance	13.4117	9.71133	15.3557
	Similarity	96.04%	96.68%	94.79%
Total Similarity		93.51%	96.89%	89.80%

4.2.3. Comparison in Unregistered Case

We designed an experiment to illustrate a significant advantage of the proposed similarity index: it pays more attention to the shape and structure of spatial distribution than traditional methods, therefore, it can better represent the spatial features of data. We simulated a misregistered case of a water body in CNLUCC data. Cases 1–3 offset 1 to 20 pixels on the x-axis, y-axis, x- and y-axis, respectively (Figure 10), and then we calculated according to the similarity index, kappa coefficient and intersection over union (IoU). The results are shown in Table 8.



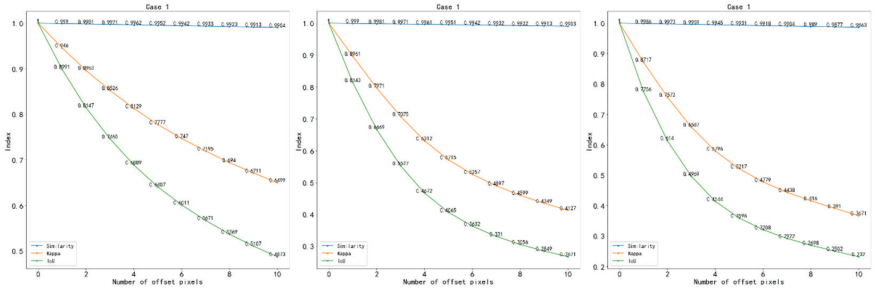
(**a**) Case 1 (**b**) Case 2 (**c**) Case 3

Figure 10. Unregistered cases (Case 1: *x*-axis; Case 2: *y*-axis; Case 3: both).

Table 8. Similarity, Kappa and IoU results of three unregistered cases.

Offset Pixel	Case 1			Case 2			Case 3		
	Similarity	Kappa	IoU	Similarity	Kappa	IoU	Similarity	Kappa	IoU
1	0.9990	0.9460	0.8991	0.999	0.8961	0.8143	0.9986	0.8717	0.7756
2	0.9981	0.8963	0.8147	0.9981	0.7971	0.6669	0.9973	0.7572	0.6140
3	0.9971	0.8526	0.7465	0.9971	0.7075	0.5527	0.9959	0.6587	0.4969
4	0.9962	0.8129	0.6889	0.9961	0.6312	0.4672	0.9945	0.5796	0.4144
5	0.9952	0.7777	0.6407	0.9951	0.5715	0.4065	0.9931	0.5217	0.3596
6	0.9942	0.7470	0.6011	0.9942	0.5257	0.3632	0.9918	0.4779	0.3208
7	0.9933	0.7195	0.5671	0.9932	0.4897	0.3310	0.9904	0.4438	0.2922
8	0.9923	0.6940	0.5369	0.9922	0.4599	0.3056	0.989	0.416	0.2698
9	0.9913	0.6711	0.5107	0.9913	0.4349	0.2849	0.9877	0.391	0.2502
10	0.9904	0.6499	0.4873	0.9903	0.4127	0.2671	0.9863	0.3671	0.2320

It could be seen that traditional methods are very sensitive to registration accuracy (shown in Figure 11).



(**a**) Case 1 (**b**) Case 2 (**c**) Case 3

Figure 11. Line charts of three indicators changes with pixel offset.

Both the kappa coefficient and IoU are not robust in three cases, because these methods do not consider spatial features of pixels; therefore, their results are quite different in cases of different pixel offsets. The similarity index proposed in this paper focuses on the structure and characteristics of space, thus, the max-sliced Wasserstein distance is close to the true pixel offset and the change of similarity result is small.

5. Discussion

In the above experiments, test datasets are used to illustrate the rationality of Wasserstein distance for quantitatively calculating the distribution of pixels in two-dimensional space. This index can comprehensively consider spatial location information and geometric shape information, which has practical significance.

After unifying the spatial resolution and classification standards of three chosen land cover products, single-category similarity and overall similarity are both calculated. In the calculation of water body similarity, the distribution of large-area water objects in products GLC_FCS30 and FROM_GLC is relatively consistent, and the difference in the surrounding scattered and small water bodies is the main reason for the generation of the distance measurement. The overall similarity results indicate that the GLC_FCS30 and CNLUCC data have a higher similarity. This is because, in FROM_GLC, the confusion between Cropland and Forest/Grass types appears more frequently, resulting in a large gap in the number of pixels between these two types. Therefore, after calculating the overall similarity with the number of pixels and similarity in the similarity evaluation index, the overall similarity is quite different from the other two products. The most similar type is Impervious surfaces, and the pixels of this type account for a large proportion of the total number of pixels, therefore, it is the main factor that affects the final similarity.

Compared with the error matrix and IoU, the similarity index represents more characteristics of spatial structure. We hope to propose this index to make up for the shortcomings of existing methods. It can better characterize the spatial relationship between datasets and solve the comparison problem under a certain degree of misregistration.

6. Conclusions

With the increasing demand for quantitative evaluation of different remote sensing products, the traditional pixel-based accuracy evaluation system needs to be improved. The proposed similarity index is a promising way to facilitate this need. It is not limited to pixel-by-pixel comparison in the case of complete alignment. In a word, the main idea of this study is to transform the relative accuracy comparison problem of multi-source raster datasets into an optimal transmission problem in two-dimensional space. For the mathematical expression of similarity index, based on max-sliced Wasserstein distance, when the number of projections reaches the threshold, it can be regarded as the true value. Using the gradient descent method to give the continuous transformation process also shows that the index can reasonably quantify the spatial distribution. Based on this similarity index, the spatial difference between multi-source raster datasets can be quantified. Theoretically, this index is more suitable for land-use change monitoring and continuous raster datasets comparison. More theoretical development and practical application of this index are still under work, and we will continue to improve it.

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

Funding: This research was funded by the National Key R&D Program, grant number No. 2019YFE0126400.

Data Availability Statement: This statement if the study did not report any data.

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

References

1. Hansen, M.C.; Reed, B. A comparison of the IGBP DISCover and University of Maryland 1 km global land cover products. *Int. J. Remote Sens.* **2000**, *21*, 1365–1373. [CrossRef]
2. Zhu, Z.; Waller, E. Global forest cover mapping for the United Nations Food and Agriculture Organization forest resources assessment 2000 program. *For. Sci.* **2003**, *49*, 369–380.

3. Stehman, S.V. Sampling designs for accuracy assessment of land cover. *Int. J. Remote Sens.* **2009**, *30*, 5243–5272. [CrossRef]
4. Visser, H.; Nijs, T.D. The Map Comparison Kit. *Environ. Model. Softw.* **2006**, *21*, 346–358. [CrossRef]
5. Ma, Z.; Redmond, R.L. Tau coefficients for accuracy assessment of classification of remote sensing data. *Photogramm. Eng. Remote Sens.* **1995**, *61*, 435–439.
6. Pontius, R., Jr.; Huffaker, D.; Denman, K. Useful techniques of validation for spatially explicit land-change models. *Ecol. Model.* **2004**, *179*, 445–461. [CrossRef]
7. Rees, W.G. Comparing the spatial content of thematic maps. *Int. J. Remote Sens.* **2008**, *29*, 3833–3844. [CrossRef]
8. Wu, B.; Zhang, L.; Zhao, Y. Feature selection via Cramer's V-test discretization for remote-sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 2593–2606. [CrossRef]
9. Ohana-Levi, N.; Gao, F.; Knipper, K.; Kustas, W.P.; Anderson, M.C.; del Mar Alsina, M.; Sanchez, L.A.; Karnieli, A. Time-series clustering of remote sensing retrievals for defining management zones in a vineyard. *Irrig. Sci.* **2021**, 1–15. [CrossRef]
10. Hagen-Zanker, A. Map comparison methods that simultaneously address overlap and structure. *J. Geogr. Syst.* **2006**, *8*, 165–185. [CrossRef]
11. San-Miguel-Ayanz, G.H. Conventional and fuzzy comparisons of large scale land cover products: Application to CORINE, GLC2000, MODIS and GlobCover in Europe. *ISPRS J. Photogramm. Remote Sens.* **2012**, *74*, 185–201.
12. Dou, W.; Ren, Y.; Wu, Q.; Ruan, S.; Chen, Y.; Bloyet, D.; Constans, J.M. Fuzzy kappa for the agreement measure of fuzzy classifications. *Neurocomputing* **2007**, *70*, 726–734. [CrossRef]
13. Hargrove, W.W.; Hoffman, F.M.; Hessburg, P.F. Mapcurves: A quantitative method for comparing categorical maps. *J. Geogr. Syst.* **2006**, *8*, 187. [CrossRef]
14. White, R. Pattern based map comparisons. *J. Geogr. Syst.* **2006**, *8*, 145–164. [CrossRef]
15. Zhu, D.; Chen, T.; Wang, Z.; Niu, R. Detecting ecological spatial-temporal changes by remote sensing ecological index with local adaptability. *J. Environ. Manag.* **2021**, *299*, 113655. [CrossRef]
16. Giri, C.; Zhu, Z.; Reed, B. A comparative analysis of the Global Land Cover 2000 and MODIS land cover data sets. *Remote Sens. Environ.* **2005**, *94*, 123–132. [CrossRef]
17. Strahler, A.H.; Boschetti, L.; Foody, G.M.; Friedl, M.A.; Hansen, M.C.; Herold, M.; Mayaux, P.; Morisette, J.T.; Stehman, S.V.; Woodcock, C.E. Global Land Cover Validation: Recommendations for Evaluation and Accuracy Assessment of Global Land Cover Maps. *Eur. Communities Luxemb.* **2006**, *51*, 1–60.
18. Foody, G.M. Assessing the accuracy of land cover change with imperfect ground reference data. *Remote Sens. Environ.* **2010**, *114*, 2271–2285. [CrossRef]
19. Foody, G.M. Explaining the unsuitability of the kappa coefficient in the assessment and comparison of the accuracy of thematic maps obtained by image classification. *Remote Sens. Environ.* **2020**, *239*, 111630. [CrossRef]
20. Enøe, C.; Georgiadis, M.P.; Johnson, W.O. Estimation of sensitivity and specificity of diagnostic tests and disease prevalence when the true disease state is unknown. *Prev. Vet. Med.* **2000**, *45*, 61–81. [CrossRef]
21. Wu, X.; Naegeli, K.; Wunderle, S. Geometric accuracy assessment of coarse-resolution satellite datasets: A study based on AVHRR GAC data at the sub-pixel level. *Earth Syst. Sci. Data* **2020**, *12*, 539–553. [CrossRef]
22. Wang, H.; Skau, E.; Krim, H.; Cervone, G. Fusing heterogeneous data: A case for remote sensing and social media. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6956–6968. [CrossRef]
23. Tardy, B.; Inglada, J.; Michel, J. Assessment of optimal transport for operational land-cover mapping using high-resolution satellite images time series without reference data of the mapping period. *Remote Sens.* **2019**, *11*, 1047. [CrossRef]
24. Deshpande, I.; Hu, Y.T.; Sun, R.; Pyrros, A.; Schwing, A. Max-Sliced Wasserstein Distance and its use for GANs. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop, Seoul, Korea, 27–28 October 2019; pp. 10648–10656.
25. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved training of wasserstein gans. *arXiv Prepr.* **2017**, arXiv:1704.00028.
26. Kolouri, S.; Rohde, G.K.; Hoffmann, H. Sliced wasserstein distance for learning gaussian mixture models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3427–3436.
27. Kolouri, S.; Park, S.R.; Rohde, G.K. The radon cumulative distribution transform and its application to image classification. *IEEE Trans. Image Processing* **2015**, *25*, 920–934. [CrossRef]
28. Zhang, X.; Liu, L.; Chen, X.; Gao, Y.; Xie, S.; Mi, J. GLC_FCS30: Global land-cover product with fine classification system at 30 m using time-series Landsat imagery. *Earth Syst. Sci. Data* **2021**, *13*, 2753–2776. [CrossRef]
29. Gong, P.; Wang, J.; Yu, L.; Zhao, Y.; Chen, J. Finer resolution observation and monitoring of global land cover: First mapping results with Landsat TM and ETM+ data. *Int. J. Remote Sens.* **2013**, *34*, 48. [CrossRef]
30. Xu, X.; Liu, J.; Zhang, S.; Li, R.; Yan, C.; Wu, S. *China's Multi-Period Land Use Land Cover Remote Sensing Monitoring Data Set (CNLUCC)*; Resource and Environment Data Cloud Platform: Beijing, China, 2018.

remote sensing

MDPI

Article

Noise Robust High-Speed Motion Compensation for ISAR Imaging Based on Parametric Minimum Entropy Optimization

Jiadong Wang [1], Yachao Li [2,*], Ming Song [3], Pingping Huang [4,5] and Mengdao Xing [2]

[1] Academy of Advanced Interdisciplinary Research, Xidian University, Xi'an 710071, China; jiadongwang@xidian.edu.cn
[2] National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China; xmd@xidian.edu.cn
[3] Beijing Institute of Space Long March Vehicle, Beijing 100097, China; songm1127@163.com
[4] College of Information Engineering, Inner Mongolia University of Technology, Hohhot 010051, China; hwangpp@imut.edu.cn
[5] Inner Mongolia Key Laboratory of Radar Technology and Application, Hohhot 010051, China
* Correspondence: ycli@mail.xidian.edu.cn

Abstract: When a target is moving at high-speed, its high-resolution range profile (HRRP) will be stretched by the high-order phase error caused by the high velocity. In this case, the inverse synthetic aperture radar (ISAR) image would be seriously blurred. To obtain a well-focused ISAR image, the phase error induced by target velocity should be compensated. This article exploits the variation continuity of a high-speed moving target's velocity and proposes a noise-robust high-speed motion compensation algorithm for ISAR imaging. The target's velocity within a coherent processing interval (CPI) is modeled as a high-order polynomial based on which a parametric high-speed motion compensation signal model is developed. The entropy of the ISAR image after high-speed motion compensation is treated as an evaluation metric, and a parametric minimum entropy optimization model is established to estimate the velocity and compensate it simultaneously. A gradient-based solver of this optimization is then adopted to iteratively find the optimal solution. Finally, the high-order phase error caused by the target's high-speed motion can be iteratively compensated, and a well-focused ISAR image can be obtained. Extensive simulation experiments have verified the noise robustness and effectiveness of the proposed algorithm.

Keywords: inverse synthetic aperture radar; space targets; high-speed motion compensation; entropy minimization; quasi-Newton iterative; noise robust

Citation: Wang, J.; Li, Y.; Song, M.; Huang, P.; Xing, M. Noise Robust High-Speed Motion Compensation for ISAR Imaging Based on Parametric Minimum Entropy Optimization. *Remote Sens.* **2022**, *14*, 2178. https://doi.org/10.3390/rs14092178

Academic Editors: Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Received: 22 March 2022
Accepted: 28 April 2022
Published: 1 May 2022

1. Introduction

Inverse synthetic aperture radar (ISAR) imaging plays an important role in space target surveillance due to its long-range, all-day, all-weather, and two-dimensional high-resolution imaging capability [1,2]. In general, the slant high-range resolution depends on transmitting wide-band linear frequency modulation (LFM) waveforms with a large pulse width. In contrast, the high azimuth resolution depends on the relative motion between the radar and the observed targets over the coherent processing interval (CPI) [3,4]. When the target is stationary or its velocity is low, the "stop–go" model is used to analyze the target echo signals [5,6]. After de-chirp processing on the receiver [7,8], the range profile of the signal can be directly extracted from the pulse compression by the Fast Fourier transform (FFT). However, for a target moving at high speed (such as missiles and satellites), their high-resolution range profile (HRRP) would be stretched by the high-order phase error induced by target velocity [9–11]. Velocity estimation and compensation are of particular significance for the ISAR imaging of high-speed moving targets, which deteriorates the quality of HRRP and ISAR image [12,13]. Therefore, to not affect the subsequent ISAR imaging processing, it is necessary to estimate the target speed and compensate for the phase error caused by the high-speed motion.

The key to the high-speed motion compensation of ISAR imaging lies in accurately estimating the target's high velocity. The current methods are divided into two main categories. One category is the algorithms based on signal decomposition to estimate the high-order phase error parameters directly for individual echo and then directly acquire the velocity for high-speed motion compensation. The fractional Fourier transform (FrFT) [14,15] and its modification [16] have been utilized to reconstruct HRRP for high-speed moving targets. Furthermore, many parameter estimation methods, such as the integrated cubic phase function (ICPF) [17,18], particle swarm optimization [19], Wigner Ville Distribution (WVD) and Hough Transform (HT) [20], etc., were utilized to estimate the target velocity from the quadratic phase error. These methods model the radar echo after pulse compression as the sum of multiple signals containing quadratic phase error, then compensate the high-speed motion by estimating the signal chirp term that contains the target velocity. This category of method relies on an accurate estimation of the signal's chirp term, which is susceptible to noise. Another class of methods uses the focusing quality of the HRRP as a criterion for the indirect estimation of velocity, and the most typical criterion is the waveform entropy [11,21,22]. Entropy is an effective metric for evaluating the focusing quality of HRRP and is used in many ISAR imaging applications such as translational motion compensation [23–25] and image auto-focusing processing [26–29]. This class of methods creates a higher-order compensation term for each compressed echo and establishes a parametric model for individual echo high-speed motion compensation. Then, the phase error is searched and compensated by minimizing the waveform entropy. The problem with these algorithms is that they treat each pulse independently, ignoring the continuity and the integrity of a high-speed moving target's motions during continuous observation. Due to the separate processing of the echoes, each echo's high-speed motion estimation error gradually accumulates within a CPI, resulting in an inefficient high-speed compensation of the image as a whole. In addition, the signal-to-noise ratio (SNR) of the echo is often relatively low for targets due to the signal decay from the long range and absorption of the transmitting medium. The SNR problem is among the most significant challenges that ISAR imaging systems frequently face. In the presence of low SNR, the high-speed motion compensation always encounters some difficulties [17,30,31]. As a result, the imaging results would degrade seriously.

Aiming to perform the ISAR imaging of a high-speed moving target, this paper proposes a noise-robust high-speed motion compensation method for ISAR imaging based on parametric minimum entropy optimization. Firstly, for the radar echoes of high-speed moving targets in the De-chirp mode [7], we analyze the influence of the high-speed motion of the target on the compressed echoes and establish the signal model for the high-speed moving target. In general, for a continuously observed target, its movement state, including its trajectory and velocity, changes in a continuous manner [25,32–34]. Considering the variation continuity of the target's velocity within one CPI, the target's velocity is modeled as a high-order polynomial function, and 2D image entropy is minimized to optimize the velocity polynomial coefficients. A novel coordinate descent algorithm is proposed to solve the minimum entropy optimization based on the established minimum entropy optimization model. The coordinate descent algorithm is implemented by the Broyden–Fletcher–Goldfarb–Shanno (BFGS)-based quasi-Newton algorithm [35–37] yields fast convergence. The effectiveness of the high-speed compensation algorithm is verified by simulation data and Yak-42-measured data. Compared with existing algorithms, the proposed method is innovative in the following aspects:

(1) The most significant advantage of the proposed method is that it considers the correlation of velocity variations of sequential echoes during one CPI. The continuity and completeness of the target velocity variation are exploited to establish the high order polynomial of the sequential echo's velocity for the high-speed motion compensation. Compared with the high-speed motion compensation method based on independent echo, the proposed method is more robust and has higher compensation accuracy.

(2) Most of the existing high-speed compensation methods use exhaustive search or signal parameter estimation, which is computationally expensive and sensitive to noise. In contrast, the proposed method uses the 2D ISAR image's entropy as an evaluation index. This uses the BFGS algorithm, which is an effective quasi-Newton algorithm that does not need to calculate the second-order derivative of the objective function. The operational speed of the BFGS algorithm is faster than that of the Newton method. That is to say, the proposed algorithm is more effective and practical.

(3) The existing high-speed motion compensation methods do not take full advantage of the high accumulation gain of sequential echoes. The proposed method can achieve high SNR gain from 2D coherent integration [33,34], which benefits the high-speed motion compensation under low SNR.

This paper is organized as follows. Section 2 presents the De-chirped signal model for high-speed moving targets. In Section 3, a parametric model of the high-speed motion compensation within one CPI is established. The minimum entropy optimization is developed, and the gradient-based solver of this optimization problem is introduced. In Section 4, some imaging results based on the simulated and measured data are given, and the performance of the proposed high-speed compensation method is analyzed. Some conclusions are given in Section 5.

2. De-Chirp Signal Model for High-Speed Moving Targets

A general geometry of the radar and target is given in Figure 1, in which a coordinate is built on the center gravity O of the target with the Y axis along the direction of LOS. In Figure 1, the plane consisting of the XY axis including the line of radar sight (LOS) is the imaging plane. The final ISAR image is the projection of the 3D target structure on the imaging plane. In radar imaging, the high range resolution is usually achieved by transmitting large band-width linear-frequency-modulated (LFM) signals with pulse compression. Assuming the radar transmits a chirp waveform that

$$s(t_r, t_m) = \mathrm{rect}\left(\frac{t_r}{T_p}\right) \cdot \exp\left[j2\pi\left(f_c t + \frac{1}{2}\gamma t_r^2\right)\right], \tag{1}$$

where $\mathrm{rect}(t_r/T_p) = \begin{cases} 1, |t_r| \le T_p/2 \\ 0, |t_r| > T_p/2 \end{cases}$, and T_p, f_c, and γ denote the pulse-width, carrier frequency, and frequency modulation rate, respectively. $t = t_r + t_m$ is the full time, where t_r is the fast time and t_m is the slow time. As shown in Figure 1, the point p is an arbitrary scatterer on the target whose distance from the radar at t_m is $R_d(t_m)$; then the radar echo of this scatter can be written as

$$s_p(t_r, t_m) = \sigma_p \mathrm{rect}\left(\frac{t_r - t_d}{T_p}\right) \cdot \exp\left[j2\pi\left(f_c(t - t_d) + \frac{1}{2}\gamma(t_r - t_d)^2\right)\right], \tag{2}$$

where $t_d = \frac{2R_d(t_m)}{c}$ is the echo time delay of point p, c is the velocity of light, σ_p is the reflection coefficient. Note the instantaneous distance from the radar to scatter p, i.e., $R_d(t_m)$ is only related to slow time t_m because a "stop–go" assumption is adopted, i.e., the radar target is supposed to move between radar pulses and stop within each pulse, as shown in Figure 2a. Noting the pulse width of the wide-band signal is generally narrow, e.g., 100 us, the "stop–go" assumption is reasonable and has been widely used in ISAR imaging. For the target moving with high velocity, however, the target movement within a pulse cannot be ignored and the assumption of "stop-go" is invalid. For example, assuming that the radar transmits an LFM signal with a bandwidth of 1 GHz and a pulse width of 100 us, for a slow-moving target with a speed of 100 m/s (such as an airplane), the distance variation within a pulse is 0.01 m. Furthermore, for a high-speed moving target with a speed of 3000 m/s (such as the satellite), the distance variation within a pulse is 0.3 m. Compared to the range resolution $\Delta r = c/2B = 0.15$ m, the distance variation within the pulse for the slow-moving target can be ignored, while it can not be neglected for the high-speed

moving target. For high-speed moving targets, since it is necessary to consider the target distance variation within one pulse-width, the distance between point p and the radar is the variable concerned with both fast time t_r and slow time t_m, which can be expressed as

$$R_d(t_r, t_m) = R_{d1}(t_m) + R_{d2}(t_r), \tag{3}$$

where $R_{d1}(t_m)$ is the distance variation with slow time, and $R_{d2}(t_r)$ is the distance variation with the fast time. Considering the fact that a pulse time is short and the change of velocity within a pulse time can be neglected, i.e., the target can be approximated to be moving at a uniform speed within a pulse, then $R_{d2}(t_r)$ can be approximated as

$$R_{d2}(t_r) \approx v(t_m) \cdot t_r, \tag{4}$$

where $v(t_m)$ is the instantaneous velocity of the target at slow time t_m. The de-chirp compression processing is expressed as the echo signal multiples with the reference signal's conjugate [7]. The reference signal is

$$s_{ref}(t_r, t_m) = \mathrm{rect}\left(\frac{t_r - t_{ref}}{T_p}\right) \exp\left[j2\pi\left(f_c\left(t - t_{ref}\right) + \frac{1}{2}\gamma\left(t_r - t_{ref}\right)^2\right)\right], \tag{5}$$

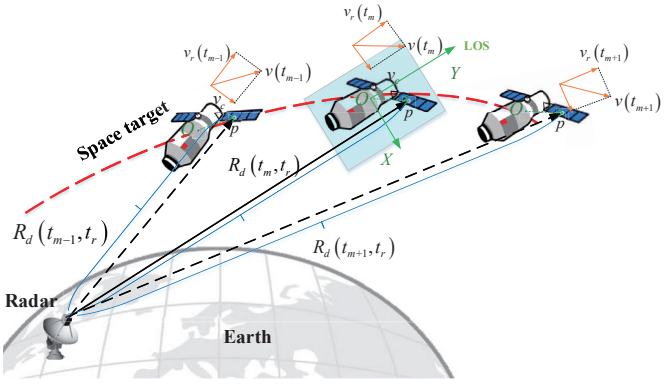


Figure 1. Observation geometry for high-speed motion targets.

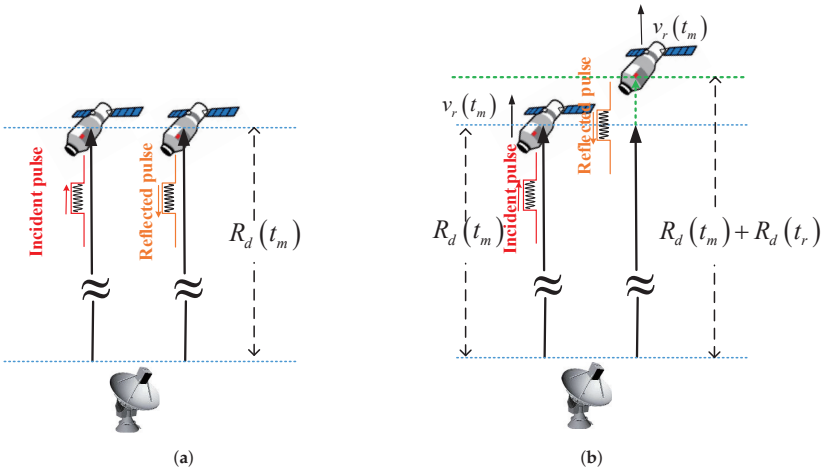


Figure 2. Difference between low-speed moving target and high-speed moving target: (**a**) the low-speed moving target; and (**b**) the high-speed moving target.

where $t_{ref} = \frac{2R_{ref}(t_m)}{c}$, $R_{ref}(t_m)$ is the reference distance from point p to the radar at slow time t_m. After the de-chirp processing, we can obtain the output signal

$$\begin{aligned} s_o(t_r, t_m) =& s_p(t_r, t_m) \cdot s_{ref}^*(t_r, t_m) \\ =& \sigma_p \text{rect}\left(\frac{t_r - t_d}{T_p}\right) \cdot \text{rect}\left(\frac{t_r - t_{ref}}{T_p}\right) \exp\left\{-j2\pi\left[f_c\left(t_d - t_{ref}\right) + \gamma t_r\left(t_d - t_{ref}\right) - \frac{1}{2}\gamma\left(t_d^2 - t_{ref}^2\right)\right]\right\}. \end{aligned} \tag{6}$$

Simplifying Equation (6) yields

$$s_o(t_r, t_m) = \sigma_p \text{rect}\left(\frac{t_r - t_d}{T_p}\right) \cdot \text{rect}\left(\frac{t_r - t_{ref}}{T_p}\right) \cdot \exp\left[-j2\pi\left(a_0 + a_1 t_r + a_2 t_r^2\right)\right], \tag{7}$$

where

$$\begin{cases} a_0 = f_c \frac{2\left[R_d(t_r, t_m) - R_{ref}(t_m)\right]}{c} - \gamma \frac{2\left[R_d^2(t_r, t_m) - R_{ref}^2(t_m)\right]}{c^2} \\ a_1 = -f_c \frac{2v(t_m)}{c} + \gamma \frac{2\left[R_d(t_r, t_m) - R_{ref}(t_m)\right]}{c} + \gamma \frac{4R_d(t_r, t_m)v(t_m)}{c^2}, \\ a_2 = \gamma \frac{2v(t_m)}{c} - \gamma \frac{2v^2(t_m)}{c^2} \end{cases} \tag{8}$$

where a_2 is the chirp term caused by the high-speed motion of the target. In ISAR imaging, the target's motion can be divided into translational and rotational motion [6], respectively, as shown in Figure 3. Assuming that the coordinate of the point p in the imaging plane XOY is (x_p, y_p), the instantaneous distance from scattering point p to radar is given by

$$\begin{aligned} R_d(t_r, t_m) &= R_o(t_r, t_m) + x_p \sin(\omega t_m) + y_p \cos(\omega t_m) + v(t_m) \cdot t_r \\ &\approx R_o(t_r, t_m) + x_p \omega t_m + y_p + v(t_m) \cdot t_r, \end{aligned} \tag{9}$$

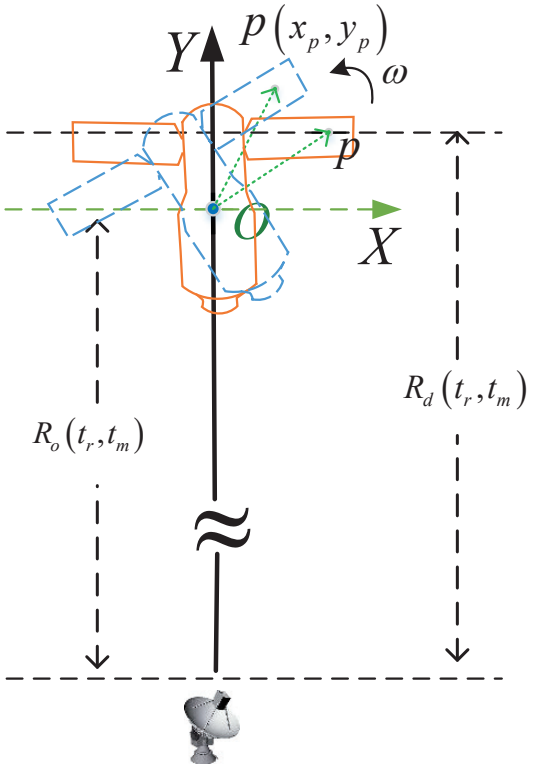


Figure 3. Target's rotational motion in ISAR imaging.

where $R_o(t_r, t_m)$ denotes the translational motion of the target, as shown in Figure 3. In ISAR imaging, since the time of a CPI is very short, such as a few milliseconds, the rotation of the target relative to the radar within a CPI is a few degrees (approximately 3°), and the target can approximate uniform rotation in a short time. At this time, the terms $\sin(\omega t_m)$ and $\cos(\omega t_m)$ in Equation (9) satisfy $\begin{cases} \sin(\omega t_m) \approx \omega t_m \\ \cos(\omega t_m) \approx 1 \end{cases}$, where ω is the rotational velocity of the target in the imaging plane. Bringing Equation (9) into Equation (7) yields

$$\begin{aligned} s_o(t_r, t_m) = \sigma_p \mathrm{rect}\left(\frac{t_r - t_d}{T_p}\right) \cdot \mathrm{rect}\left(\frac{t_r - t_{ref}}{T_p}\right) \cdot \exp\left[-j\frac{4\pi}{\lambda} x_p \omega t_m\right] \\ \cdot \exp\left[-j\frac{4\pi}{c}\gamma t_r y_p\right] \cdot \exp\left[-j\frac{4\pi}{\lambda} y_p\right] \cdot \exp\left[-j\frac{4\pi}{c}\gamma x_p \omega t_m t_r\right] \\ \cdot \exp\left[-j\frac{4\pi}{\lambda}\Big(R_o(t_m) - R_{ref}(t_m)\Big)\right] \cdot \exp\left[-j\frac{4\pi}{c}\frac{\gamma\Big[R_d^2(t_m) - R_{ref}^2(t_m)\Big]}{c}\right] \\ \cdot \exp\left[-j\frac{4\pi}{c}\left(\gamma\Big(R_o(t_m) - R_{ref}(t_m)\Big) - f_c v(t_m) + \gamma\frac{2R_d(t_m)v(t_m)}{c}\right)t_r\right] \\ \cdot \exp\left[-j\frac{4\pi}{c}\gamma\left(v(t_m) - \frac{v^2(t_m)}{c}\right)t_r^2\right], \end{aligned} \tag{10}$$

where $\lambda = \frac{c}{f_c}$ is the wavelength, and the phase in Equation (10) is divided into eight terms. The first term is the rotational Doppler term of point p and the second term is the range compression term of point p. These two terms are the time domain data corresponding to the final image of the target. The third term is constant and can be ignored. The fourth term is the range walk term due to rotational motion, which usually does not exceed one range cell in ISAR imaging, whose effect can be neglected [38,39]. The fifth term is the phase error from the translational movement of the target as a whole, which can be removed by the autofocus algorithm [6,40,41]. The 6th term is the residual video phase (RVP) error, which can be removed by RVP compensation. The seventh term is the envelope linear walk term brought by the target translational and high-speed motion, which can be eliminated by envelope alignment [23,42]. The eighth term is the range chirp term brought by the high-speed movement of the target, which needs to be compensated in this paper. After the envelope alignment [23] and phase error compensation [40], Equation (10) becomes

$$s_o(t_r, t_m) \approx \tilde{s}(t_r, t_m) \cdot \exp\left[-j4\pi\gamma\left(\frac{v(t_m)}{c} - \frac{v^2(t_m)}{c^2}\right)t_r^2\right], \tag{11}$$

where $\tilde{s}(t_r, t_m)$ is the time domain data of the ideal image after high-speed compensation, denoted as

$$\tilde{s}(t_r, t_m) = \sigma_p \mathrm{rect}\left(\frac{t_r - t_d}{T_p}\right) \cdot \mathrm{rect}\left(\frac{t_r - t_{ref}}{T_p}\right) \exp\left[-j\frac{4\pi}{\lambda} x_p \omega t_m\right] exp\left[-j\frac{4\pi}{c} y_p \gamma t_r\right]. \tag{12}$$

According to Equation (11), the high-speed compensation signal model for ISAR imaging can be expressed as

$$\tilde{s}(t_r, t_m) \approx s_o(t_r, t_m) \cdot \exp\left[j4\pi\gamma\left(\frac{v(t_m)}{c} - \frac{v^2(t_m)}{c^2}\right)t_r^2\right]. \tag{13}$$

Applying the fast Fourier transform (FFT) with respect to t_r and t_m and considering the inevitable noise, Equation (13) can be expressed in a discrete form as

$$\begin{aligned} g(k,h) = & \sum_{m}^{M-1} \exp\left(-j2\pi\frac{hm}{M}\right) \sum_{n=0}^{N-1} \exp\left(-j2\pi\frac{kn}{N}\right) \\ & \cdot s_o(n,m) \exp\left[j4\pi\gamma\left(\frac{v(m)}{c} - \frac{v^2(m)}{c^2}\right)n^2\right] + \xi(k,h), \end{aligned} \tag{14}$$

where $g(k,h)$ is the ISAR image after high-speed motion compensation. $k = 1, 2, \cdots, N$, k is the range indices, N is the number of range bins, and $h = 1, 2, \cdots, M$, where h is the azimuth position and M is the number of azimuth cells. $s_o(n,m)$ is the discrete form of $s_o(t_r, t_m)$, n and m are the discrete form of t_r and t_m, $\xi(k,h)$ denotes the complex noise. Equation (14) is the signal model of the final ISAR images after high-speed motion compensation. In the following sections, the parametric minimum entropy optimized high-speed motion compensation algorithm is given based on this signal model.

3. Optimal Compensation for High-Speed Motion

3.1. Optimization Based on Parametric Minimum Entropy

From Equation (14), it can be seen that the velocity of the target varies with the slow time t_m. The high-speed compensation for independent echoes does not consider the continuity of velocity variation [15,20,21]. Due to the complex motion of the target and the variance of the system and the environment, the high-velocity between the target and the radar usually has high-order terms [24,25,33,34]. Without loss of generality, we model the target's high-velocity as an L-order polynomial, meaning that

$$v(m) = \sum_{l=0}^{L-1} b_l (m\Delta t_m)^l, \tag{15}$$

where l denotes the order of velocity variation with slow time t_m, $l = 0, 1, \cdots, L-1$, and b_l represents the coefficient of each order. Δt_m denotes the pulse repetition time (PRT). One notes that l begins from 0 to $L-1$, b_0 indicates the initial value of the velocity. For simplicity and clarity, we define the polynomial coefficient vector as $\mathbf{b} = [b_0, b_1, b_2, \cdots, b_{L-1}]_{1\times L}$, and give the complex image after error correction by the high-speed compensation term as follows:

$$\begin{aligned} g(k,h) = & \sum_{m}^{M-1} \exp\left(-j2\pi\frac{hm}{M}\right) \sum_{n=0}^{N-1} \exp\left(-j2\pi\frac{kn}{N}\right) \\ & \cdot s_o(n,m) \exp\left[j4\pi\gamma\left(\frac{\sum_{l=0}^{L-1} b_l (m\Delta t_m)^l}{c} - \frac{\left(\sum_{l=0}^{L-1} b_l (m\Delta t_m)^l\right)^2}{c^2}\right)n^2\right] + \xi(k,h). \end{aligned} \tag{16}$$

If the values of $\mathbf{b} = [b_0, b_1, b_2, \cdots, b_{L-1}]_{1\times L}$ are obtained precisely, the high-speed motion of the target will be compensated, and a well-focused ISAR image can be obtained. Therefore, the high-speed motion compensation problem turns into an optimal parameter estimation problem. Actually, estimating the optimal parameters set in $v(m) = \sum_{l=0}^{L-1} b_l (m\Delta t_m)^l$ can be transferred into solving an unconstrained optimization problem in which $\mathbf{b} = [b_0, b_1, b_2, \cdots, b_{L-1}]_{1\times L}$ are the variables of objective function.

Image entropy [25–27] and contrast [37,41] are frequently used in ISAR imaging to quantify the image focus. In this paper, image entropy is employed to evaluate the focus quality of images. Entropy has been used as a typical indicator in ISAR imaging in many different ways [28,29]. The entropy of the 2-D image represents its sharpness, and generally,

the "sharpest" image corresponds to the entirely focused image. The complex image after high-speed motion correction by $\tilde{\mathbf{b}} = \left[\tilde{b}_0, \tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_{L-1}\right]_{1\times L}$ can be rewritten as

$$g(k,h;\tilde{\mathbf{b}}) = \sum_{m}^{M-1} \exp\left(-j2\pi\frac{hm}{M}\right) \sum_{n=0}^{N-1} \exp\left(-j2\pi\frac{kn}{N}\right) \cdot s_o(n,m) \exp\left[j4\pi\gamma\left(\frac{\sum_{l=0}^{L-1}\tilde{b}_l(m\Delta t_m)^l}{c} - \frac{\left(\sum_{l=0}^{L-1}\tilde{b}_l(m\Delta t_m)^l\right)^2}{c^2}\right)n^2\right] + \xi(k,h). \tag{17}$$

Therefore, the entropy of an image is defined as a function of $\tilde{\mathbf{b}} = \left[\tilde{b}_0, \tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_{L-1}\right]_{1\times L}$, which is given by

$$E_g(\tilde{\mathbf{b}}) = \ln S_g - \frac{1}{S_g}\sum_{k=0}^{N-1}\sum_{h=0}^{M-1}\left|g(k,h;\tilde{\mathbf{b}})\right|^2 \ln\left|g(k,h;\tilde{\mathbf{b}})\right|^2, \tag{18}$$

where S_g is the image intensity that can be expressed as

$$S_g = \sum_{k=0}^{N-1}\sum_{h=0}^{M-1}\left|g(k,h;\tilde{\mathbf{b}})\right|^2. \tag{19}$$

The estimate of $\tilde{\mathbf{b}} = \left[\tilde{b}_0, \tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_{L-1}\right]_{1\times L}$ is obtained by minimizing the image entropy, expressed as follows:

$$\left\langle \hat{b}_0, \cdots, \hat{b}_{L-1} \right\rangle = \arg \min_{\tilde{b}_0,\cdots,\tilde{b}_{L-1}} E_g(\tilde{\mathbf{b}}). \tag{20}$$

To date, the optimization based on entropy minimization for high-speed motion compensation is established, and it is an optimization function with series parameters $\tilde{\mathbf{b}} = \left[\tilde{b}_0, \tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_{L-1}\right]_{1\times L}$. Many standard algorithms are available to solve this optimization, such as particle swarm optimization (PSO) and genetic algorithms (GA)[19]. However, these nonparameteric methods always need great computation time. BFGS is an effective quasi-Newton algorithm that efficiently solves unconstrained optimization problems. In the following subsection, we present a fast iterative optimization search method based on the BFGS quasi-Newton iteration method [37].

3.2. Parameter Optimization Based on Fast Iteration

To apply the BFGS-based fast iterative search method, one first has to obtain the gradient of each parameter. For an arbitrary parameter $\tilde{b}_{l_0}$, $l_0 \in [0, 1, \cdots, L-1]$, its gradient is

$$\frac{\partial E_g(\tilde{\mathbf{b}})}{\partial \tilde{b}_{l_0}} = -\frac{1}{S_g}\sum_{k=0}^{N-1}\sum_{h=0}^{M-1}\left\{\left[1 + \ln\left|g(k,h;\tilde{\mathbf{b}})\right|^2\right] \cdot \frac{\partial\left|g(k,h;\tilde{\mathbf{b}})\right|^2}{\partial \tilde{b}_{l_0}}\right\}, \tag{21}$$

where $\left|g(k,h;\tilde{\mathbf{b}})\right|^2 = g(k,h;\tilde{\mathbf{b}}) \cdot g^*(k,h;\tilde{\mathbf{b}})$; then, we have

$$\frac{\partial\left|g(k,h;\tilde{\mathbf{b}})\right|^2}{\partial \tilde{b}_{l_0}} = 2\mathrm{Re}\left[g^*(k,h;\tilde{\mathbf{b}}) \cdot \frac{\partial g(k,h;\tilde{\mathbf{b}})}{\partial \tilde{b}_{l_0}}\right], \tag{22}$$

where

$$\begin{aligned}\frac{\partial g(k,h;\tilde{\mathbf{b}})}{\partial \tilde{b}_{l_0}} = & \sum_{m}^{M-1} \exp\left(-j2\pi\frac{hm}{M}\right) \sum_{n=0}^{N-1} \exp\left(-j2\pi\frac{kn}{N}\right) \\ & \cdot s_o(n,m) \cdot \exp\left[j4\pi\gamma\left(\frac{\sum_{l=0}^{L-1}\tilde{b}_l(m\Delta t_m)^l}{c} - \frac{\left(\sum_{l=0}^{L-1}\tilde{b}_l(m\Delta t_m)^l\right)^2}{c^2}\right)n^2\right] \\ & \cdot \left[j4\pi\gamma\left(\frac{1}{c} - \frac{2\sum_{l=0}^{L-1}\tilde{b}_l(m\Delta t_m)^l}{c^2}\right)(m\Delta t_m)^{l_0}\right],\end{aligned} \tag{23}$$

With the partial derivative expressions in Equation (23), the gradient of image entropy with respect to $\tilde{\mathbf{b}} = \left[\tilde{b}_0, \tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_{L-1}\right]_{1\times L}$ is

$$\nabla \mathbf{E}_g(\tilde{\mathbf{b}}) = \left[\frac{\partial E_g(\tilde{\mathbf{b}})}{\partial \tilde{b}_0}, \frac{\partial E_g(\tilde{\mathbf{b}})}{\partial \tilde{b}_1}, \cdots, \frac{\partial E_g(\tilde{\mathbf{b}})}{\partial \tilde{b}_{L-1}}\right]^T. \tag{24}$$

In the BFGS algorithm, an approximate matrix (defined as $\mathbf{B}$, whose initial form is $\mathbf{B}^0 = \mathbf{I}_{L\times L}$), is used to replace the Hessian matrix of the objective function. In this paper, since we are searching for each polynomial parameter individually, $\mathbf{B}^0 = 1$. In addition, considering the considerable number of velocity polynomial parameters, it is difficult to ensure the algorithm's convergence speed and convergence robustness if the joint iterative search is performed for all parameters simultaneously. Therefore, to improve the convergence speed while providing the algorithm's robustness, a BFGS-based quasi-Newton coordinate descent algorithm was used in this paper. Herein, this paper minimized the entropy $E_g(\tilde{\mathbf{b}})$ with respect to a single parameter while holding the other parameter constant to avoid the local optimum. For example, for the parameter $\tilde{b}_{l_0}$, with the first l_0 parameters $\tilde{b}_0 \sim \tilde{b}_{l_0-1}$ which are already iteratively updated, the minimum entropy optimization function of $\tilde{b}_{l_0}$ can be expressed as

$$\left\langle \hat{b}_{l_0} \right\rangle = \arg\min_{\tilde{b}_{l_0}} E_g(\tilde{\mathbf{b}})\Big|_{\tilde{b}_{0,\cdots l_0-1, l_0+1,\cdots, L-1}=0}. \tag{25}$$

In the coordinate descent iterative algorithm, each parameter $\tilde{b}_{l_0}$ is solved independently iteratively and optimally. Considering that in Equation (25), $\tilde{b}_l = 0$, $l = 0, \cdots, l_0-1, l_0+1, \cdots, L-1$. Taking this into Equation (23), the gradient of the independent parameter $\tilde{b}_{l_0}$ can be expressed as

$$\begin{aligned}\frac{\partial g(k,h;\tilde{b}_{l_0})}{\partial \tilde{b}_{l_0}} = & \sum_{m}^{M-1} \exp\left(-j2\pi\frac{hm}{M}\right) \sum_{n=0}^{N-1} \exp\left(-j2\pi\frac{kn}{N}\right) \\ & \cdot s_o(n,m) \cdot \exp\left[j4\pi\gamma\left(\frac{\tilde{b}_{l_0}(m\Delta t_m)^{l_0}}{c} - \frac{\left(\tilde{b}_{l_0}(m\Delta t_m)^{l_0}\right)^2}{c^2}\right)n^2\right] \\ & \cdot \left[j4\pi\gamma\left(\frac{1}{c} - \frac{2\tilde{b}_{l_0}(m\Delta t_m)^{l_0}}{c^2}\right)(m\Delta t_m)^{l_0}\right].\end{aligned} \tag{26}$$

For each parameter $\tilde{b}_{l_0}$, its iterative solution process is based on the BFGS algorithm. Let $\tilde{b}_{l_0}^0$ be the initial parameter and $\tilde{b}_{l_0}^k$ be the parameter of the *kth* iteration. The searching direction in BFGS is updated as follows:

$$\mathbf{d}^k = -\mathbf{B}^k \cdot \nabla \mathbf{E}_g\left(\tilde{b}_{l_0}^k\right). \tag{27}$$

The $k+1th$ parameter $\tilde{b}_{l_0}^{k+1}$ is updated as follows:

$$\tilde{b}_{l_0}^{k+1} = \tilde{b}_{l_0}^k + \lambda^k \mathbf{d}^k, \tag{28}$$

where λ^k is the search step corresponding to $\tilde{b}_{l_0}^k$ at the *kth* iteration. It can be estimated by Equation (29) via some 1-D inexact searching methods, such as golden section search or the Armijo–Goldstein stepsize rule [37].

$$\lambda^k = \arg\min_{\lambda_k}\left[E_g\left(\tilde{b}_{l_0}^k + \lambda^k \mathbf{d}^k\right)\right]. \tag{29}$$

The Hessian matrix $\mathbf{B}^k$ in BFGS is updated as follows:

$$\mathbf{B}^{k+1} = \mathbf{B}^k + \frac{\mathbf{y}^k \cdot \left(\mathbf{y}^k\right)^T}{\mathbf{y}^k \cdot \left(\mathbf{s}^k\right)^T} - \frac{\mathbf{B}^k \mathbf{s}^k \left(\mathbf{s}^k\right)^T \mathbf{B}^k}{\left(\mathbf{s}^k\right)^T \mathbf{B}^k \mathbf{s}^k}, \tag{30}$$

where $\mathbf{s}^k = \lambda^k d^k$, $\mathbf{y}^k = \nabla \mathbf{E}_g\left(\tilde{b}_{l_0}^{k+1}\right) - \nabla \mathbf{E}_g\left(\tilde{b}_{l_0}^k\right)$.

All parameters are updated throughout the parameter optimization process in two loop iterations, the inner and outer loops, respectively. Within the inner loop, for the parameter $\tilde{b}_l, l = 0, 1, \cdots, L-1$, the parameters are independently updated based on BFGS from $\tilde{b}_0 \sim \tilde{b}_{L-1}$ in turn, and each parameter is independently updated as an inner loop. When all parameters are updated once, it is an outer loop, and after completing an outer loop, it goes to a new outer loop. Until the image entropy satisfies a certain value, the iteration stops.

To clearly describe the proposed algorithm, a flowchart of the proposed algorithm is given in Figure 4.

As can be seen from the flow chart, first, the polynomial order is selected. Since the time of a CPI is very short, say less than 1 second, the target's velocity variation is small, and a velocity polynomial of order 1–2 can accurately describe the high-speed motion of the target. In this paper, L is set to 5 to make the proposed algorithm more robust, i.e., it can satisfy the case of weak target maneuver as well as the case of strong target maneuver. For $L = 5$, the algorithm only sacrifices a small amount of computation time, but this will make the high-speed compensation more accurate and robust. The iterative process is divided into an inner loop and an outer loop. The inner loop is a BFGS-based gradient search for each polynomial parameter independently. After a complete search estimation of all order coefficients, the range alignment and phase adjustment were re-implemented. This process is an outer loop, where δ_1 and δ_2 are the inner loop and outer loop iteration termination conditions, respectively. In general, $\delta_1 = 10^{-3}$, $\delta_2 = 10^{-4}$ are usually a good choice in reality. It is important to emphasize that after all the coefficients are updated, the range profile of each pulse will be changed because the high-speed motion of the target was compensated to a certain extent. Hence, it is necessary to realign all the echo envelopes [23] and refocus the image [26,27] for the next iteration.

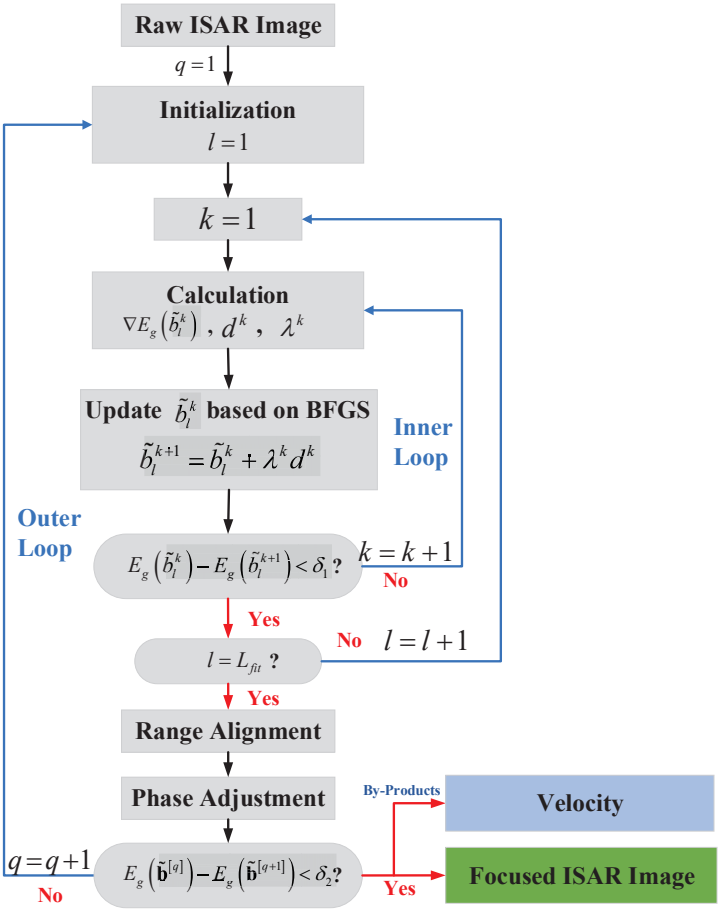


Figure 4. The flow chart of the proposed algorithm.

4. Experiment Analysis

To verify the effectiveness of the proposed algorithm, in this subsection, different experiments were designed to demonstrate the performance of the proposed algorithm. The experiments are divided into the three types and all the images are generated by the conventional range-Doppler (RD) [1] imaging algorithm. The difference is that different high-speed motion compensation algorithms are used. For all experiments, the proposed method is compared with the algorithm in [21], which uses the idea of minimum entropy of individual echoes for high-speed compensation. It is referred to as ME. The proposed algorithm is also compared with the algorithm proposed in [17], which uses ICPF to estimate the chirp coefficients of independent echoes and thus for high-speed compensation. It is referred to as ICPF in this paper. It is important to emphasize that before the high-speed motion compensation, the translation compensation [23] and the phase error compensation [40] are applied to compensate for the fifth, sixth, and seventh terms in Equation (10).

(1) Firstly, point simulation experiments are designed to verify the performance of the proposed algorithm under different high-speed motion conditions.

(2) Considering the difficulty of obtaining the real measurement data of the space target, this paper uses the electromagnetic simulation data of the space target for the experiment, and the PO algorithm [43] obtains simulation data. In addition, to illustrate

the robustness of the proposed algorithm in the case of low SNR, the performance of the proposed algorithm under different SNR conditions is given in the experiments.

(3) In this part of the experiment, different high-speed motion was added to the Yak-42 real measured data to evaluate the effectiveness of the proposed method, and the high-speed motion was added using Equation (11).

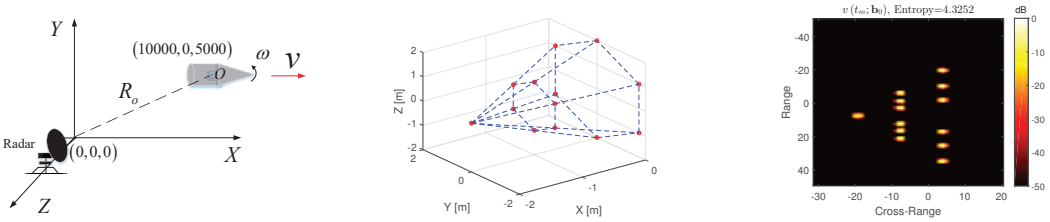
4.1. Experiments Based on Point Array Simulation

The first experiment is based on scattering point simulation. A simulated ballistic missile consisting of 13 scatterers is constructed as shown in Figure 5b, which is supposed to fly straight above the radar with different projected velocities. The motion model is given in Figure 5a. The radar transmits a linear frequency modulation (LFM) signal with the parameters given in Table 1. The signal-to-noise ratio (SNR) of the radar echo is 20 dB. The radar echo simulation was carried out under different high-speed motion conditions, as shown in Table 2. In this paper, the SNR of a signal is defined as

$$\mathrm{SNR} = 10\log_{10}\left(\frac{E_s}{E_n}\right), \tag{31}$$

where E_s denotes the energy of the radar echo, and E_n denotes the energy of white Gaussian noise.

The high-speed compensated ISAR imaging results under different motion conditions are shown in Figure 6, all the imaging results were obtained using the RD imaging algorithm [1]. The left column of Figure 6 is the ISAR images without high-speed motion compensation. It can be seen that, as the target speed increases, the high-speed motion also has an increasing impact on the ISAR imaging results, and the images become increasingly blurred. The second column of Figure 6 shows the ISAR images acquired by the ME. It can be seen that the focusing image quality is significantly improved by using the high-speed compensation algorithm. The third column of Figure 6 shows the ISAR images acquired by the ICPF. It can be seen that the image focusing quality obtained by ICPF is basically the same as that of the ME algorithm. Compared with uncompensated images, the scattered points in the images are well focused. The fourth column of Figure 6 shows the focused ISAR images acquired by the proposed algorithm, and it can be seen that the proposed algorithm achieves images with better focusing quality. For comparison, the entropy of the images after high-speed compensation by different algorithms are given, as shown in Table 3. As can be seen from the table, compared with the ME and ICPF, the image entropy obtained by the proposed algorithm is smaller and closer to the ideal image. The variation of the image entropy with the iteration number of the proposed algorithm is given in the right column of Figure 6, and it can be seen that the proposed algorithm reaches convergence after approximately ten iterations. Considering that the first and second-order of the target velocity dominate the influence within a CPI, the image entropy against the velocity and acceleration are given in Figure 7a–d. It can be seen that the adoption of the global image entropy as the evaluation criterion has a global minimum, and the algorithm can robustly converge to the global optimum.



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

Figure 5. (**a**) Target movement trajectory; (**b**) the scattering point model; and (**c**) the ideal image of zero velocity.

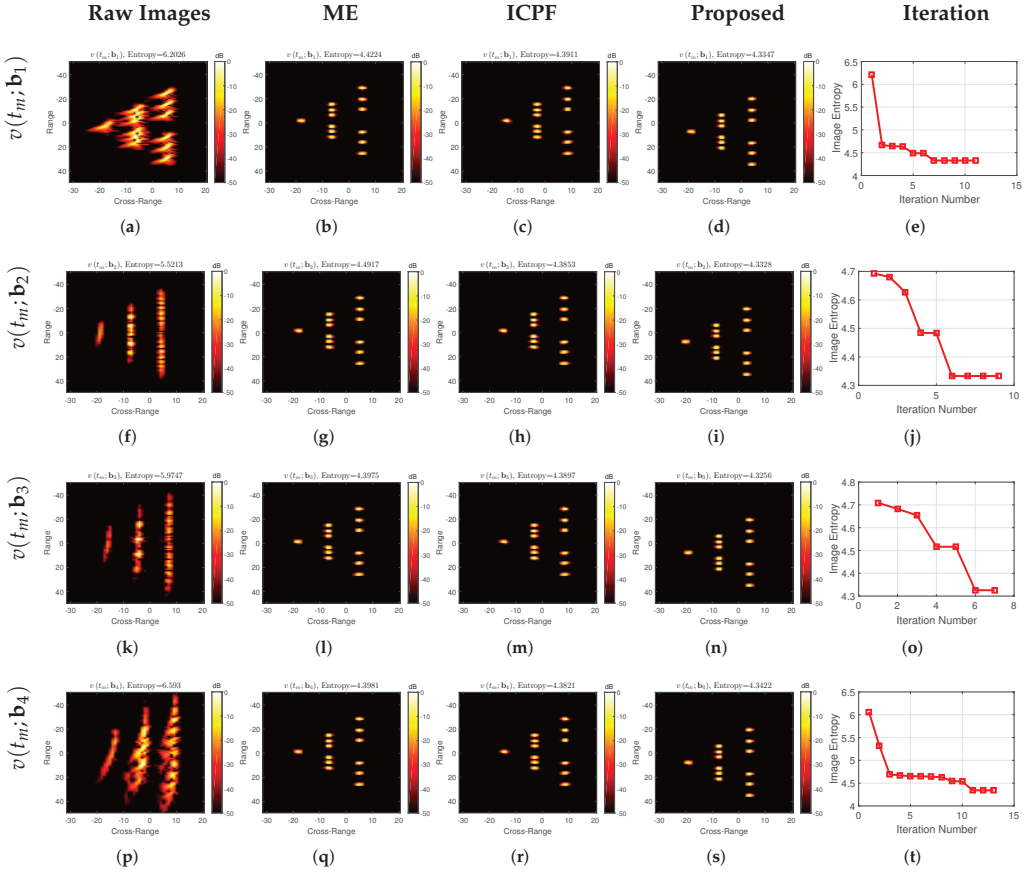


Figure 6. High-speed compensated imaging results under different motion conditions. The leftmost column is the imaging results without high-speed motion compensation; the second column is the high-speed compensation imaging results by ME; the third column is the high-speed compensation imaging results by ICPF; the fourth column is the high-speed compensation imaging results of the proposed algorithm; the rightmost column is the image entropy against the iteration number of the proposed algorithm.

Table 1. Radar parameters of the simulation.

Center Frequency	Pulse Repetition Frequency	Pulse Width	Band Width	Sample Frequency
16 GHz	1000 Hz	400 us	2 GHz	10 MHz

Table 2. Motion parameters for the simulation.

v \ $\mathbf{b}$	b_0	b_1	b_2
$v(t_m;\mathbf{b}_0)$	0	0	0
$v(t_m;\mathbf{b}_1)$	1000	1000	10
$v(t_m;\mathbf{b}_2)$	3000	1000	100
$v(t_m;\mathbf{b}_3)$	5000	2000	1000
$v(t_m;\mathbf{b}_4)$	7000	6000	10

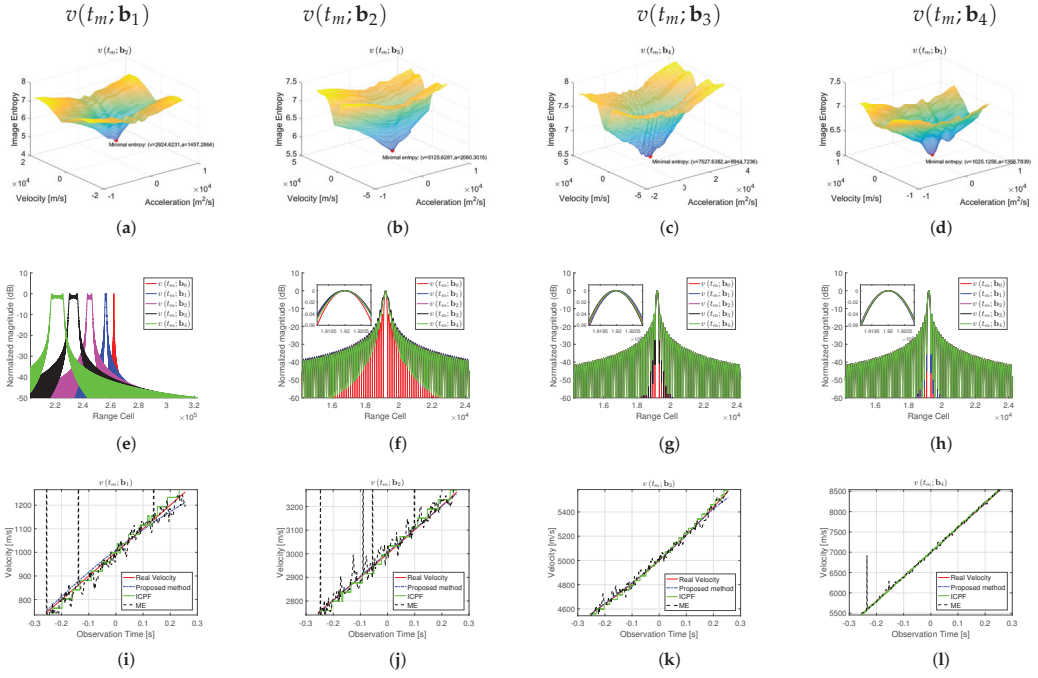


Figure 7. (**a**–**d**) Image entropy variation with velocity and acceleration; (**e**–**h**) range profiles of individual scattering points in ISAR images; and (**i**–**l**) velocity estimation and its true value of different high-speed compensation algorithms at different motion conditions.

Table 3. The entropy of images acquired by different algorithms.

Image Entropy				
Ideal Image	**4.3252**			
	$v(t_m;\mathbf{b_1})$	$v(t_m;\mathbf{b_2})$	$v(t_m;\mathbf{b_3})$	$v(t_m;\mathbf{b_4})$
Raw Image	6.2026	5.5213	5.9747	6.593
ME	4.4224	4.4917	4.3975	4.3981
ICPF	4.3911	4.3853	4.3897	4.3821
Proposed Method	4.3347	4.3328	4.3256	4.3422

To better reflect the advantages of the proposed algorithm, the range profiles of the independent scattering points of the image are given in Figure 7e–h. From Figure 7e, it can be seen that as the target velocity continues to increase, the range chirp term brought by the high-velocity becomes more and more prominent, and the profile spreading after range compression becomes more and more serious. The range profile after high-speed motion compensation is shown in Figure 7f–h, and it can be seen that after high-speed motion compensation, the main lobe broadening of the independent points disappears, forming a well-focused range compression lobe. However, compared with the proposed algorithm, the main lobe of the range profile after the compensation of ME and ICPF still has the spreading phenomenon. In contrast, after the compensation of the proposed algorithm, the main lobe has no broadening.

Figure 7i–l gives the estimated velocity of the three algorithms and the true velocity. One can see that since the ME and ICPF algorithms process each echo independently from the velocity estimation, the estimated velocity is not correlated. The velocity estimates of each pulse are independent of each other. There are many speed estimates that deviate significantly from the true value, which will eventually lead to inadequate image compensation. In contrast, the proposed algorithm considers the continuity of the target's velocity variation within a CPI, and the estimated velocity is consistent with the actual value which also reflects the effectiveness of the proposed algorithm. The root mean square error (RMES) of velocity estimated by different algorithms is shown in Table 4, and RMES is defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\tilde{v}_{estimate}(n) - v_{real}(n)\right)^2} \tag{32}$$

where $\tilde{v}_{estimate}$ is the estimated velocity and v_{real} is the true velocity. It can be seen from the RMSE that the estimation error of the proposed algorithm is much lower than the errors of the comparison methods, which proves the effectiveness of the proposed algorithm.

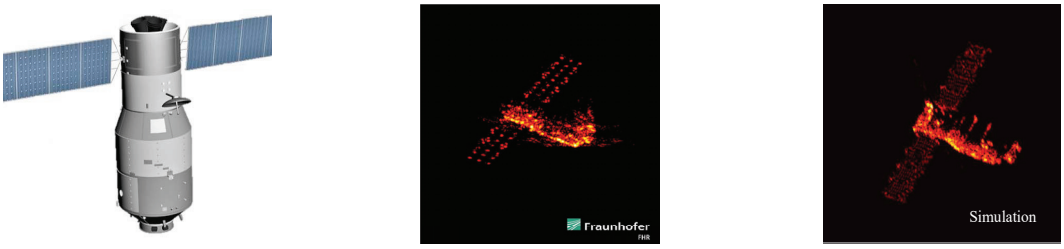
Table 4. Estimated speed RMSE of different algorithms with point simulation experiments.

RMSE				
	$v(t_m;\mathbf{b_1})$	$v(t_m;\mathbf{b_2})$	$v(t_m;\mathbf{b_3})$	$v(t_m;\mathbf{b_4})$
ME	217.41	254.57	27.27	65.99
ICPF	19.55	18.46	19.24	19.76
Proposed Method	17.35	4.40	12.83	6.46

4.2. Experiments Based on TG-I's Electromagnetic Simulation

Since satellite data are rarely publicly available, the experimental data in this subsection are obtained based on electromagnetic simulations with the electromagnetic model TG-1, whose 3D model is shown in Figure 8a. All simulations adopt a triangular facet model to divide the target surface into thousands of equivalent scatterers. The radar echo

data of a solid object are calculated by adopting the fast physical optics (FPO) algorithm [1], and the conventional RD algorithm generates the ISAR images for EM simulation. To illustrate the validity of the EM simulation, a comparison between the real ISAR image of TG-I (Figure 8a) and the EM simulation ISAR image (Figure 8b) is given in Figure 8. The German FGAN Lab published the measurement image of TG-I in March 2018 (at Fraunhofer FHR, available at https://www.fhr.fraunhofer.de/tiangong-bilder; accessed on 21 March 2018). The comparison result shows that the performance of the generated imagery is close to that of the measured ISAR image, which supports the investigation in this article. The radar parameters and target's motion parameters used for the simulation are the same as the experiments in the previous section.



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

Figure 8. (**a**) The CAD model of TG-I satellite; (**b**) real ISAR image; and (**c**) EM simulation ISAR image.

First, the imaging results of the different algorithms for high-speed compensation under different motion conditions are given, as shown in Figure 9. The left column of Figure 9 shows the imaging results without high-speed motion compensation, where one can see that as the target speed increases, the blurring of the ISAR images becomes increasingly severe, and the entropy value of the images becomes larger. When looking at the two high-speed compensation algorithms, since the electromagnetic simulation is closer to the actual measurement data than the previous simple scattering point simulation, the high-speed compensation of ME and ICPF is not satisfactory. The focusing quality of the images is minimally improved. In contrast, the algorithm proposed in this paper can still accurately compensate, and the image after high-speed compensation can be accurately focused, reflecting the robustness of the proposed algorithm. For comparison, the entropy of the images after high-speed compensation by different algorithms are given, as shown in Table 5. As can be seen from the table, compared with the ME and ICPF, the image entropy obtained by the proposed algorithm is smaller and closer to the ideal image. From Figure 10, it can be seen that the estimated velocity using ME and ICPF have a significant error, and the bias between the estimated velocity and the true value can reach several kilometers per second, which is the main reason for the failure of the algorithm. In contrast, the estimated velocity of the proposed algorithm basically matches the true value, and the error is basically negligible, which reflects the effectiveness of the proposed algorithm. Similarly, the RMSE for the speed estimation of different algorithms is given, as shown in Table 6. From the table, it can be seen that the speed estimation of the proposed algorithm is more accurate and has less error.

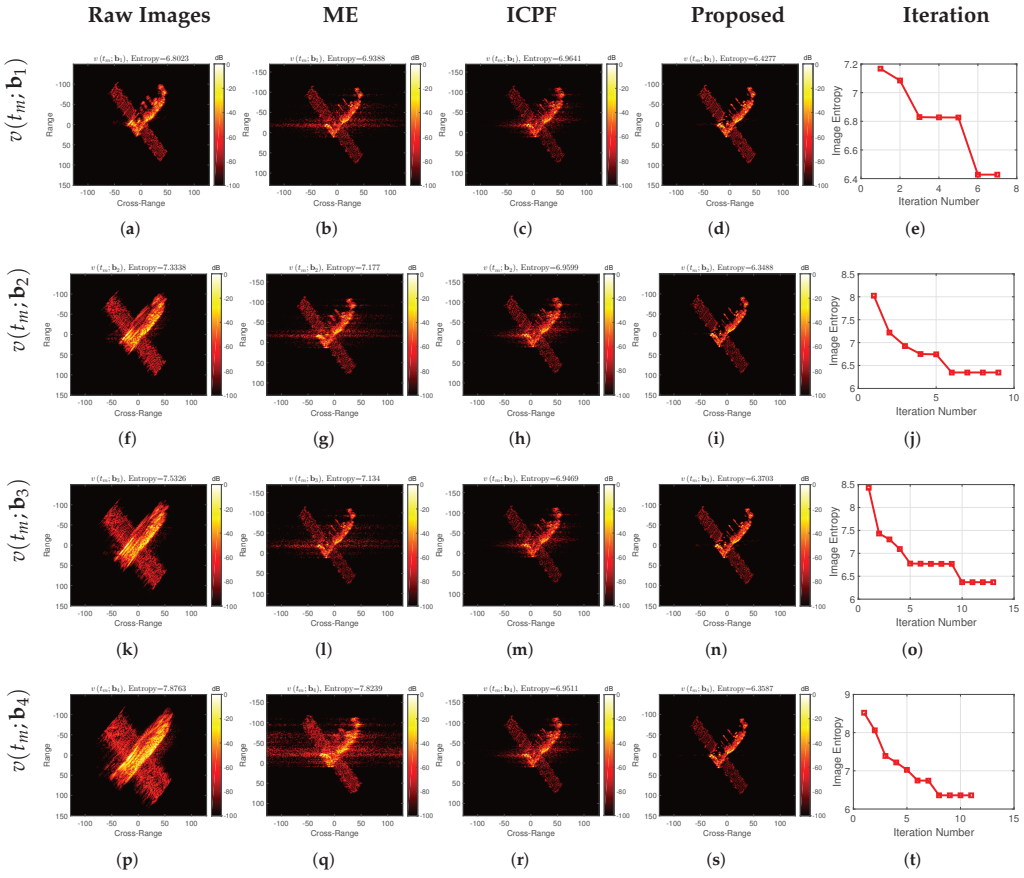


Figure 9. Experimental results of TG-I electromagnetic simulation under different motion conditions. The leftmost column shows the imaging results without high-speed motion compensation; the second column shows the high-speed compensation imaging results by ME; the third column shows the high-speed compensation imaging results by ICPF; the fourth column shows the high-speed compensation imaging results of the proposed algorithm; the rightmost column is the image entropy against the iteration number of the proposed algorithm.

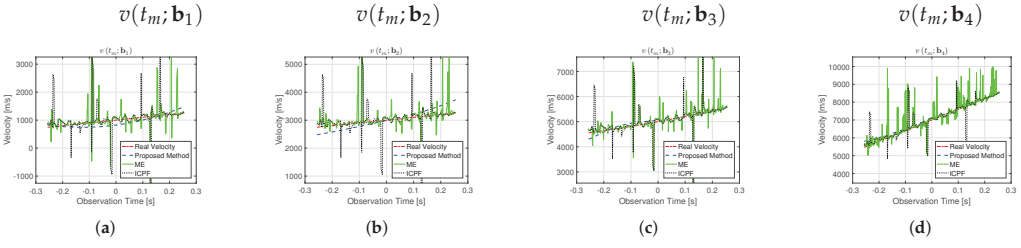


Figure 10. Comparison of the estimated velocity and real velocity using TG-I EM simulation data under different motion conditions.

Table 5. The entropy of images acquired by different algorithms using TG-I EM simulation data.

Image Entropy				
Ideal Image	**6.3441**			
	$v(t_m;\mathbf{b_1})$	$v(t_m;\mathbf{b_2})$	$v(t_m;\mathbf{b_3})$	$v(t_m;\mathbf{b_4})$
Raw Images	6.8023	7.3338	7.5326	7.8763
ME	7.181	7.177	7.134	7.8239
ICPF	6.9641	6.9599	6.9469	6.9511
Proposed Method	6.4277	6.3488	6.3703	6.3587

Table 6. Estimated speed RMSE of different algorithms using TG-I EM simulation data.

RMSE				
	$v(t_m;\mathbf{b_1})$	$v(t_m;\mathbf{b_2})$	$v(t_m;\mathbf{b_3})$	$v(t_m;\mathbf{b_4})$
ME	492.3	414.1	454.8	549.5
ICPF	503.6	503.5	503.5	457.0
Proposed Method	125.2	202.8	83.4	123.7

4.3. Performance Under Different SNRs

To verify the performance of the proposed method under low SNR, the complex white Gaussian noise is added to electromagnetic simulation data with velocity parameters of $v(t_m;\mathbf{b}_4)$ to generate different SNRs (from 0 dB to -13 dB). Figure 11 shows the experiment results under different SNRs. The images without high-speed compensation are shown in the first column in Figure 11, corresponding to the SNR equivalent of 0 dB, -5 dB, -10 dB, and -13 dB, respectively. The images obtained from the ME and ICPF are given in the second and third columns of Figure 11. The images obtained from the proposed method are given in the fourth column. Furthermore, entropy against the iteration number is shown in the last column of Figure 11. As one can note, even under the low SNR conditions, the proposed gradient-based optimization usually achieves convergence within less than 15 iterations. It is notable in Figure 11 that the images obtained without high-speed motion compensation are poor in quality due to the strong noise. It cannot generate focused images when the SNR is less than -5 dB. In addition, it can be seen that the images generated by the high-speed compensation algorithm based on ME and ICPF have some improvement in focus quality. However, in the case of low SNR, such as below -5 dB, both algorithms have failed, and it is basically impossible to focus the imaging.

In contrast, the proposed algorithm can realize the accurate compensation for high-speed target motion at -13 dB and achieve well-focused images. Table 7 gives the entropy of the high-speed compensated images for different algorithms at different SNRs. The table shows that the proposed algorithm performs the best, and the proposed algorithm obtains the smallest image entropy compared to the other algorithms.

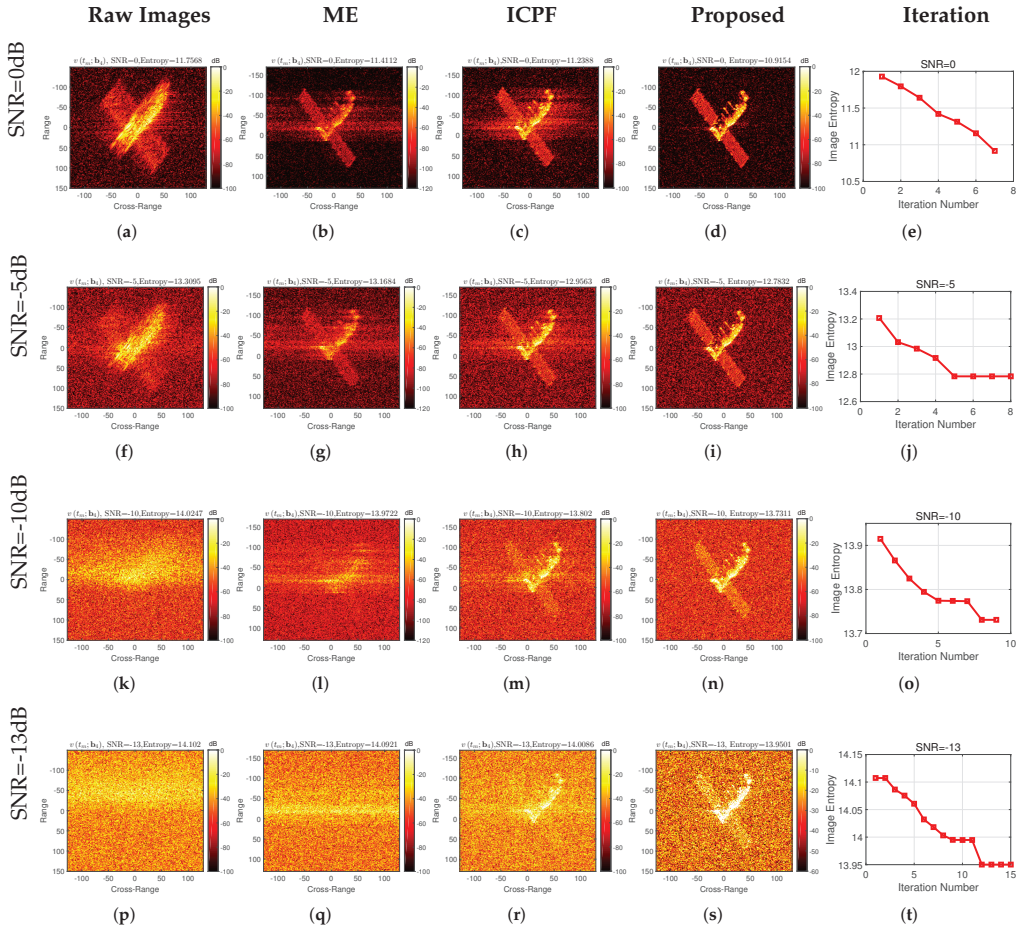


Figure 11. Experimental results of TG-I electromagnetic simulation under different SNRs.

Table 7. Entropy of high-speed compensated images with different SNRs based on TG-I's EM simulation data.

Image Entropy vs. SNR				
SNR	**0 dB**	**−5 dB**	**−10 dB**	**−13 dB**
Raw Images	11.7568	13.3095	14.0247	14.102
ME	11.4112	13.1684	13.9722	14.0921
ICPF	11.2388	12.9563	13.802	14.0086
Proposed Method	10.9154	12.7832	13.7311	13.9501

In the experiments, we found that when the SNR decreases below −13 dB, the proposed method will fail to compensate for the high-speed motion accurately, and the compensated images will be seriously blurred. To illustrate these, the RMSE curves between the estimated velocity and the true velocity for different SNRs are given, as shown in Figure 12. As one can note, the proposed method provides very small MSE only when SNR is above −13 dB, while the speed estimation errors of the other two compared methods significantly increase at SNR lower than −5 dB. When the SNR decreases below −13 dB, the RMSE of the estimated velocity becomes much more significant, which leads to blurred images. As has

been mentioned before, the relationship between the focusing quality and image entropy is inconsistent when extreme noise is involved in the data. Furthermore, the entropy of the image almost relies on the strong noise only, independent of the high-speed motion compensation. In this situation, one can use more pulses to obtain high-SNR gain, and then, the well-focused images may be generated by the proposed method. In general, the proposed algorithm has good noise robustness.

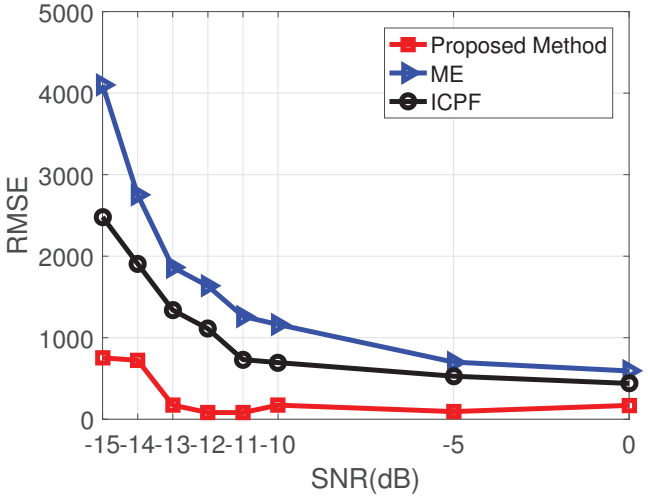


Figure 12. RMSE under different SNRs based on EM simulation data.

4.4. Experiment Using Measured Yak-42 Data

To verify the performance of the proposed algorithm on the measured data, this section uses the Yak-42 measured data for the performance analysis of the algorithm. High-speed motion and different noise are added into the data, and the different high-speed motion compensation algorithms are performed. The dataset of the Yak-42 airplane is recorded by a C-band (5.52 GHz) ISAR experimental system. The system transmits a 400 MHz linear modulated chirp signal with 25.6 us pulse duration, providing a range resolution of 0.375 m. The de-chirp sampling rate is also 10 MHz. The SNR is up to 22 dB of the raw data. The picture of the Yak-42 aircraft is shown in Figure 13a. The standard ISAR image is shown in Figure 13b. Since the speed of the actual aircraft is relatively low (approximately 100 m/s), the speed of the aircraft itself is negligible compared to the high-speed motion of several kilometers per second. In addition, different high speed motions in Table 2 are added to the original radar echoes according to Equation (10). As in the two previous experiments, the transnational motion compensation and phase error compensation are performed first, followed by the high-speed motion compensation with different algorithms.

As one can clearly see from Figure 14, compared with ME and ICPF, significantly clearer images can be achieved by using the proposed method, no matter which high-speed motions are added into the measured Yak-42 data. On the contrary, the images obtained by ME and ICPF are poor in focusing quality, although it is greatly improved compared to the images without high-speed compensation. To better show the advantage of the proposed method, Table 8 gives the image entropy after different high-speed motion compensation algorithms, as it can be seen that the proposed algorithm has the smallest image entropy after compensation, which is basically close to the entropy of the ideal image. Figure 15 gives the estimated velocity of different algorithms with respect to the real velocity, and Table 9 gives the RMSE of the estimated velocity, and it can be seen that the proposed algorithm still has the best performance on the real measured data.

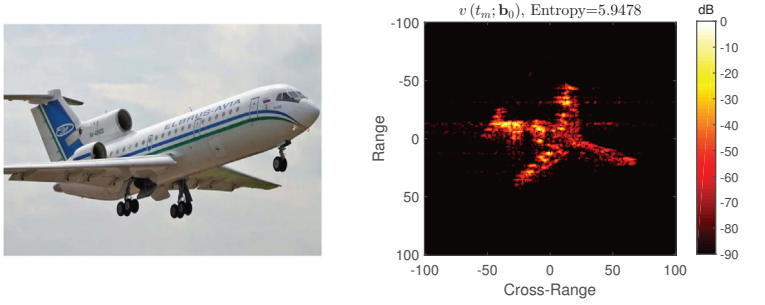


Figure 13. (**a**) Yak-42 airplane and (**b**) its standard ISAR image.

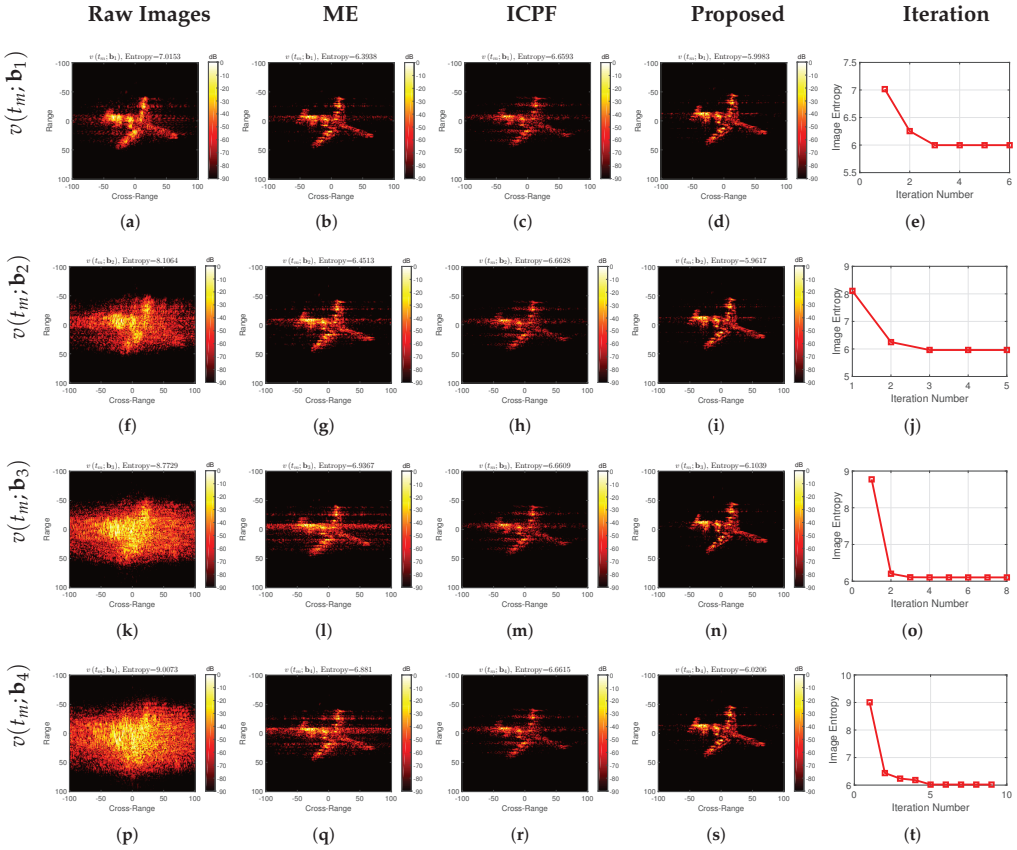


Figure 14. Experimental results of Yak-42 measured data under different motion conditions. The leftmost column is the imaging results without high-speed motion compensation; the second column is the high-speed compensation imaging results by ME; the third column is the high-speed compensation imaging results by ICPF; the fourth column is the high-speed compensation imaging results of the proposed algorithm; the rightmost column is the image entropy against the iteration number of the proposed algorithm.

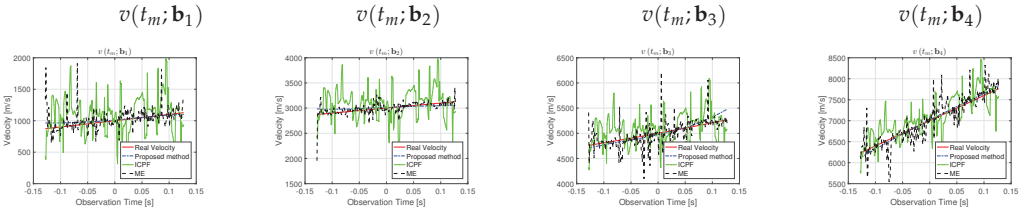


Figure 15. Comparison of estimated velocity and real velocity using Yak-42 measured data at different motion conditions.

Table 8. The entropy of images acquired by different algorithms using Yak-42 measured data.

Image Entropy				
Ideal Image	**5.9478**			
	$v(t_m;\mathbf{b}_1)$	$v(t_m;\mathbf{b}_2)$	$v(t_m;\mathbf{b}_3)$	$v(t_m;\mathbf{b}_4)$
Raw Images	7.0153	8.1064	8.7729	9.0073
ME	6.3938	6.4513	6.9367	6.881
ICPF	6.6593	6.6628	6.6609	6.6615
Proposed Method	5.9983	5.9617	6.1039	6.0206

Table 9. Estimated speed RMSE of different algorithms using Yak-42 measured data

RMSE				
	$v(t_m;\mathbf{b}_1)$	$v(t_m;\mathbf{b}_2)$	$v(t_m;\mathbf{b}_3)$	$v(t_m;\mathbf{b}_4)$
ME	146.99	93.78	194.24	252.78
ICPF	319.42	319.61	319.99	321.25
Proposed Method	35.28	49.79	66.59	49.87

The results of the high-speed motion compensation using Yak-42 measured data with different SNRs are given in Figure 16, and the different columns are the imaging results obtained by using different compensation algorithms. It can be seen that, similarly to the EM simulation data results, the proposed algorithm obtains well-focused images at low SNR (not lower than -13 dB), while both the ME and ICPF algorithms fail at low SNRs. Similarly, the entropies of the compensated images for different SNRs are given in Table 10. The RMSE of velocity estimation for different SNRs is also given, as shown in Figure 17. It can be seen that the proposed method performs the best.

To reflect the speed advantage of the proposed algorithm, a comparison of the computation time of the proposed algorithm with several other algorithms is given in Table 11. The cpu time is obtained with MATLAB coding using a personal computer with an Intel Core i5 3.30-GHz processer and 8-GB memory. From the table, it can be seen that the proposed method requires only a few seconds for the computation time, while the other two compared algorithms require several hundred seconds. This is due to the fact that the proposed method compensates all the echoes within a CPI consistently, taking into account the integrity of the target motion. However, the other two algorithms process each pulse individually and require a longer computing time.

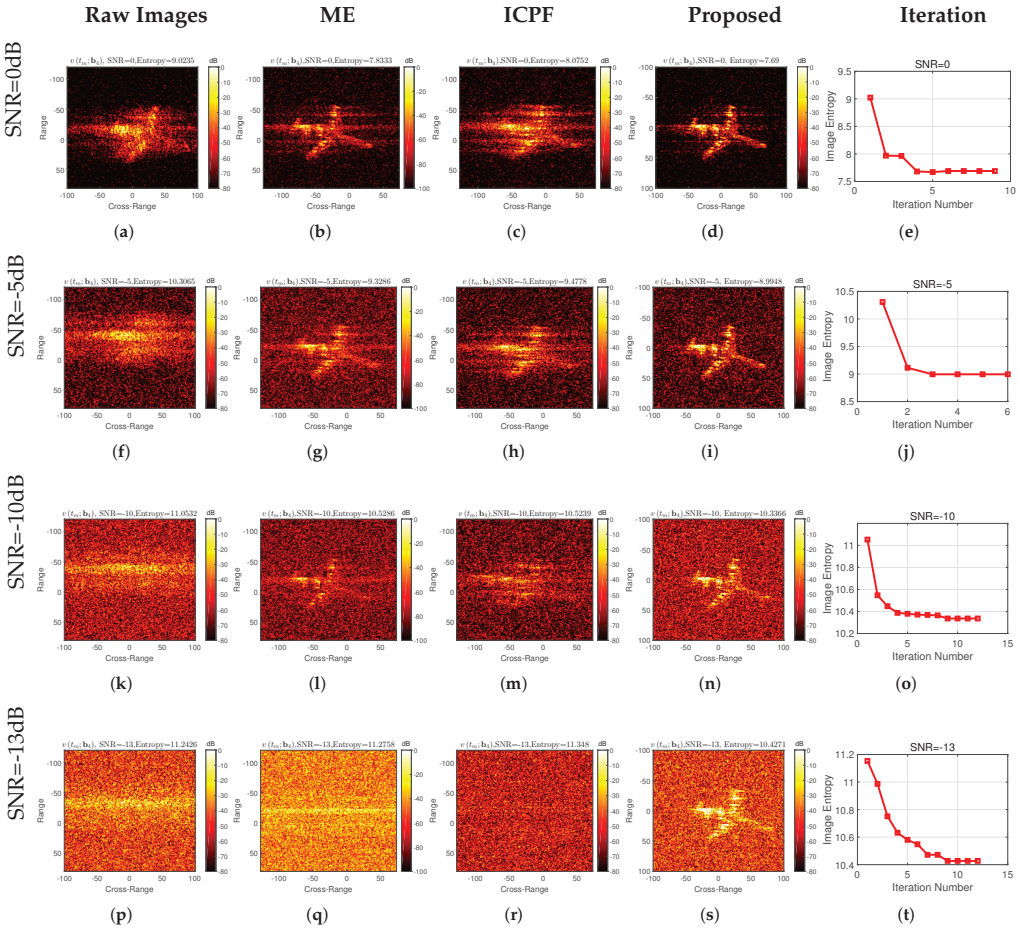


Figure 16. Experimental results of Yak-42 measured data under different SNRs.

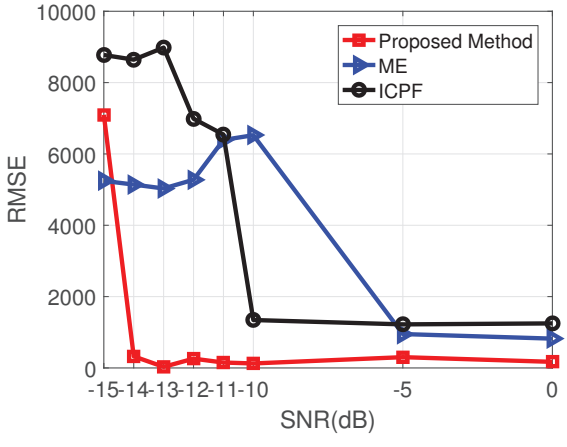


Figure 17. RMSE under different SNRs based on Yak-42 measured data.

Table 10. Entropy of high-speed compensated images with different SNRs based on Yak-42 measured data.

Image Entropy Vs SNR				
SNR	**0 dB**	**−5 dB**	**−10 dB**	**−13 dB**
Raw Images	9.0235	10.3065	11.0532	11.2426
ME	7.8333	9.3286	10.5286	11.2758
ICPF	8.0752	9.4778	10.5239	11.348
Proposed Method	7.69	8.9948	10.3366	10.4271

Table 11. Computation time comparison of individual methods.

Algorithms	**ME**	**ICPF**	**Proposed Method**
Computation time (s)	70.91	224.77	5.52

5. Conclusions

The target's high-speed motion leads to the range profile spreading after echo pulse compression, which seriously affects the ISAR imaging and leads to severe image blurring. In addition, the low SNR of the high-speed moving target echoes has been a critical problem that plagues accurate and robust high-speed motion compensation. This paper proposes a noise-robust high-speed motion compensation algorithm for the high-speed moving target ISAR imaging under low SNR conditions. This paper innovatively considers the continuity of the target velocity variation. By transforming the velocity within a CPI into a high-order polynomial model, the proposed method establishes a parameterized minimum entropy optimization model and realizes the high-speed motion compensation for the targets by quickly and accurately searching the polynomial coefficients via the BFGS-based quasi-Newton iterative method. The proposed algorithm has promising noise robustness and can accurately compensate for the high-speed motion of the target under low SNR conditions. Different experiments verify the effectiveness of the proposed algorithm.

Author Contributions: Conceptualization and methodology, J.W. and Y.L.; software, J.W.; resources, J.W. and M.S.; writing—review and editing, J.W., Y.L., P.H. and M.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key R&D Program of China under Grant 2018YFB2202500, in part by the National Natural Science Foundation of China (Grant No. 62171337), in part by the Key R&D program of Shaanxi Province under grant 2017KW-ZD-12, in part by the Shaanxi Province Funds for Distinguished Young youths under grant S2020-JC-JQ-0056, in part by the National Natural Science Foundation of China (Grant No. 62101396) and in part by the Fundamental Research Funds for the Central Universities (No. XJS212205).

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank all reviewers and editors for their comments on this paper.

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

References

1. Walker, J.L. Range-Doppler imaging of rotating objects. *IEEE Trans. Aerosp. Electron. Syst.* **1980**, *AES-16*, 23–52. [CrossRef]
2. Xu, G.; Xing, M.D.; Zhang, L.; Duan, J.; Chen, Q.Q.; Bao, Z. Sparse apertures ISAR imaging and scaling for maneuvering targets. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2942–2956. [CrossRef]
3. Yang, L.; Xing, M.; Zhang, L.; Sun, G.C.; Gao, Y.; Zhang, Z.; Bao, Z. Integration of rotation estimation and high-order compensation for ultrahigh-resolution microwave photonic isar imagery. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 2095–2115. [CrossRef]
4. Ma, J.T.; Gao, M.G.; Guo, B.F.; Dong, J.; Xiong, D.; Feng, Q. High resolution inverse synthetic aperture radar imaging of three-axis-stabilized space target by exploiting orbital and sparse priors. *Chin. Phys. B* **2017**, *26*, 108401. [CrossRef]

5. Jakowatz, C.V.; Wahl, D.E.; Eichel, P.H.; Ghiglia, D.C.; Thompson, P.A. *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach: A Signal Processing Approach*; Springer: Berlin/Heidelberg, Germany, 2012.
6. Chen, C.C.; Andrews, H.C. Target-motion-induced radar imaging. *IEEE Trans. Aerosp. Electron. Syst.* **1980**, *AES-16*, 2–14. [CrossRef]
7. Caputi, W.J. Stretch: A time-transformation technique. *IEEE Trans. Aerosp. Electron. Syst.* **1971**, *AES-7*, 269–278. [CrossRef]
8. Wehner, D.R. High resolution radar. In *Norwood*; Artech House: London, UK, 1987.
9. Tian, B.; Chen, Z.; Xu, S.; Liu, Y. ISAR imaging compensation of high speed targets based on integrated cubic phase function. In *MIPPR 2013: Multispectral Image Acquisition, Processing, and Analysis*; International Society for Optics and Photonics: Bellingham, Washington, USA, 2013; Volume 8917, p. 89170B.
10. Kun-Fan, Z.; Zhi-Hong, F.; De-Bao, M. Study on a method of compensation for the range profile of high velocity spatial targets. In Proceedings of the 2010 International Conference on Image Analysis and Signal Processing, Zhejiang, China, 9–11 April 2010; pp. 450–453.
11. Tian, B.; Lu, Z.; Liu, Y.; Li, X. High velocity motion compensation of IFDS data in ISAR imaging based on adaptive parameter adjustment of matched filter and entropy minimization. *IEEE Access* **2018**, *6*, 34272–34278. [CrossRef]
12. Gu, F.F.; Fu, M.H.; Chen, C.H.; Yang, M.; Zhang, Y. A novel ISAR imaging method for high speed moving target based on parametric sparse representation. In Proceedings of the 2017 16th International Conference on Optical Communications and Networks (ICOCN), Wuzhen, China, 7–10 August 2017; pp. 1–3.
13. Zhiping, Y.; Zhen, F.; Dongjin, W.; Weidong, C. ISAR imaging of fast-moving target based on FRFT range compression. In Proceedings of the 2007 1st Asian and Pacific Conference on Synthetic Aperture Radar, Huangshan, China, 5–9 November 2007; pp. 306–309.
14. Cao, M.; Fu, Y.; Jiang, W.; Li, X.; Zhuang, Z. High resolution range profile imaging of high speed moving targets based on fractional Fourier transform. In *MIPPR 2007: Automatic Target Recognition and Image Analysis; and Multispectral Image Acquisition*; International Society for Optics and Photonics: Bellingham, WA, USA, 2007; Volume 6786, p. 678654.
15. Zhang, S.; Sun, S.; Zhang, W.; Zong, Z.; Yeo, T.S. High-resolution bistatic ISAR image formation for high-speed and complex-motion targets. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 3520–3531. [CrossRef]
16. Wang, F.; Jiang, D.; Chen, H. High range resolution profile construction exploiting modified fractional Fourier transformation. *Math. Probl. Eng.* **2015**, *2015*, 321878. [CrossRef]
17. Liu, Y.; Zhang, S.; Zhu, D.; Li, X. A novel speed compensation method for ISAR imaging with low SNR. *Sensors* **2015**, *15*, 18402–18415. [CrossRef]
18. Wang, Y.; Kang, J.; Jiang, Y. ISAR imaging of maneuvering target based on the local polynomial Wigner distribution and integrated high-order ambiguity function for cubic phase signal model. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2971–2991. [CrossRef]
19. Brinkman, W.; Thayaparan, T. Focusing ISAR images using the AJTF optimized with the GA and the PSO algorithm-comparison and results. In Proceedings of the 2006 IEEE Conference on Radar, Verona, NY, USA, 24–27 April 2006.
20. He, C.; Daiying, Z. High speed motion compensation based on the range profile. In Proceedings of the 2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013), Kunming, China, 5–8 August 2013; pp. 1–4.
21. Sheng, J.; Fu, C.; Wang, H.; Liu, Y. High speed motion compensation for terahertz ISAR imaging. In Proceedings of the 2017 International Applied Computational Electromagnetics Society Symposium (ACES), Suzhou, China, 1–4 August 2017; pp. 1–2.
22. Guo, B.; Li, Z.; Xiao, Y.; Shi, L.; Han, N.; Zhu, X. ISAR Speed Compensation Algorithm for High-speed Moving Target Based on Simulate Anneal. In Proceedings of the 2019 IEEE 19th International Conference on Communication Technology (ICCT), Xi'an, China, 16–19 October 2019; pp. 1595–1599.
23. Zhu, D.; Wang, L.; Yu, Y.; Tao, Q.; Zhu, Z. Robust ISAR range alignment via minimizing the entropy of the average range profile. *IEEE Geosci. Remote Sens. Lett.* **2009**, *6*, 204–208.
24. Liu, L.; Zhou, F.; Tao, M.; Sun, P.; Zhang, Z. Adaptive translational motion compensation method for ISAR imaging under low SNR based on particle swarm optimization. *IEEE J.Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 5146–5157. [CrossRef]
25. Zhang, L.; Sheng, J.l.; Duan, J.; Xing, M.D.; Qiao, Z.J.; Bao, Z. Translational motion compensation for ISAR imaging under low SNR by minimum entropy. *EURASIP J. Adv. Signal Process.* **2013**, *2013*, 33. [CrossRef]
26. Xi, L.; Guosui, L.; Ni, J. Autofocusing of ISAR images based on entropy minimization. *IEEE Trans. Aerosp. Electron. Syst.* **1999**, *35*, 1240–1252. [CrossRef]
27. Wang, J.; Liu, X.; Zhou, Z. Minimum-entropy phase adjustment for ISAR. *IEE Proc.-Radar Sonar Navig.* **2004**, *151*, 203–209. [CrossRef]
28. Kragh, T.J.; Kharbouch, A.A. Monotonic iterative algorithm for minimum-entropy autofocus. In Proceedings of the Adaptive Sensor Array Processing (ASAP) Workshop, Lexington, KY, USA, 6–7 June 2006; Volume 40, pp. 1147–1159.
29. Wang, J.; Zhang, L.; Du, L.; Yang, D.; Chen, B. Noise-robust motion compensation for aerial maneuvering target ISAR imaging by parametric minimum entropy optimization. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4202–4217. [CrossRef]
30. Li, Y.; Wu, R.; Xing, M.; Bao, Z. Inverse synthetic aperture radar imaging of ship target with complex motion. *IET Radar Sonar Navig.* **2008**, *2*, 395–403. [CrossRef]

31. Liu, Y.; Li, G.; Tian, B.; Chen, Z.P. ISAR imaging at low SNR level based on polarimetric whitening filter. In *MIPPR 2013: Multispectral Image Acquisition, Processing, and Analysis*; International Society for Optics and Photonics: Bellingham, WA, USA, 2013; Volume 8917, p. 891703.
32. Barbarossa, S.; Di Lorenzo, P.; Vecchiarelli, P. Parameter estimation of 2D multi-component polynomial phase signals: An application to SAR imaging of moving targets. *IEEE Trans. Signal Process.* **2014**, *62*, 4375–4389. [CrossRef]
33. Cantoni, A.; Martorella, M. Fourier-based ISAR imaging using 2D polynomials. *IET Radar Sonar Navig.* **2017**, *11*, 1216–1227. [CrossRef]
34. Cantoni, A.; Martorella, M. ISAR image autofocus using 2D-polynomials. In Proceedings of the 2016 IEEE Radar Conference (RadarConf), Philadelphia, PA, USA, 2–6 May 2016; pp. 1–6.
35. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer: Berlin/Heidelberg, Germany, 2006.
36. Shao, S.; Zhang, L.; Liu, H.; Zhou, Y. Spatial-variant contrast maximization autofocus algorithm for ISAR imaging of maneuvering targets. *Sci. China Inf. Sci.* **2019**, *62*, 40303. [CrossRef]
37. Shao, S.; Zhang, L.; Liu, H.; Zhou, Y. Accelerated translational motion compensation with contrast maximisation optimisation algorithm for inverse synthetic aperture radar imaging. *IET Radar Sonar Navig.* **2019**, *13*, 316–325. [CrossRef]
38. Chen, V.C.; Lipps, R. ISAR imaging of small craft with roll, pitch and yaw analysis. In Proceedings of the Record of the IEEE 2000 International Radar Conference, Alexandria, VA, USA, 12 May 2000; pp. 493–498.
39. Chen, V.C.; Miceli, W. Time-varying spectral analysis for radar imaging of manoeuvring targets. *IEE Proc.-Radar Sonar Navig.* **1998**, *145*, 262–268. [CrossRef]
40. Li, Y.; Xing, M.; Su, J.; Quan, Y.; Bao, Z. A new algorithm of ISAR imaging for maneuvering targets with low SNR. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 543–557. [CrossRef]
41. Martorella, M.; Berizzi, F.; Haywood, B. Contrast maximisation based technique for 2-D ISAR autofocusing. *IEE Proc.-Radar Sonar Navig.* **2005**, *152*, 253–262. [CrossRef]
42. Wang, J.; Kasilingam, D. Global range alignment for ISAR. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 351–357. [CrossRef]
43. Boag, A. A fast physical optics (FPO) algorithm for high frequency scattering. *IEEE Trans. Antennas Propag.* **2004**, *52*, *AES-16*, 197–204. [CrossRef]

remote sensing

MDPI

Article

LiDAR Filtering in 3D Object Detection Based on Improved RANSAC

Bingxu Wang †, Jinhui Lan *,† and Jiangjiang Gao

School of Automation, University of Science and Technology Beijing, 30 Xueyuan Road, Haidian District, Beijing 100083, China; b20190278@xs.ustb.edu.cn (B.W.); g20198662@xs.ustb.edu.cn (J.G.)

* Correspondence: lanjh@ustb.edu.cn

† These authors contributed equally to this work.

Abstract: At present, the LiDAR ground filtering technology is very mature. There are fewer applications in 3D-object detection due to the limitations of filtering accuracy and efficiency. If the ground can be removed quickly and accurately, the 3D-object detection algorithm can detect objects more accurately and quickly. In order to meet the application requirements of 3D-object detection, inspired by Universal-RANSAC, we analyze the detailed steps of RANSAC and propose a precise and efficient RANSAC-based ground filtering method. The principle of GroupSAC is analyzed, and the sampled points are grouped by attributes to make it easier to sample the correct point. Based on this principle, we devise a method for limiting sampled points that is applicable to point clouds. We describe preemptive RANSAC in detail. Its breadth-first strategy is adopted to obtain the optimal plane without complex iterations. We use the International Society for Photogrammetry and Remote Sensing (ISPRS) datasets and the KITTI dataset for testing. Experiments show that our method has higher filtering accuracy and efficiency compared with the currently widely used methods. We explore the application of ground filtering methods in 3D-object detection, and the experimental results show that our method can improve the object detection accuracy without affecting the efficiency.

Keywords: light detection and ranging (LiDAR) filtering; random sample consensus (RANSAC); Universal-RANSAC; 3D-object detection

Citation: Wang, B.; Lan, J.; Gao, J. LiDAR Filtering in 3D Object Detection Based on Improved RANSAC. *Remote Sens.* **2022**, *14*, 2110. https://doi.org/10.3390/rs14092110

Academic Editors: Yue Wu, Kai Qin, Qiguang Miao and Maoguo Gong

Received: 14 March 2022
Accepted: 23 April 2022
Published: 28 April 2022

1. Introduction

Light detection and ranging (LiDAR) can obtain real three-dimensional spatial coordinate information within the measurement range [1]. This has the characteristics of high efficiency and high accuracy. LiDAR is widely used in urban planning, agricultural development, environmental monitoring, and transportation [2]. Ground filtering is a key technology to separate and extract ground information from the point-cloud data obtained by LiDAR [3,4]. Point cloud filtering is a significant step in the process of point-cloud processing [5]. Therefore, in the past two decades, scholars have proposed many effective automatic filtering algorithms, which greatly reduce the labor costs and improve the application efficiency of point-cloud data [2]. There are many widely used methods that are well suited for different situations. However, these methods can still be better optimized.

With the development of deep-learning technology [6], 3D-object-detection methods based on convolutional neural networks have achieved high accuracy and efficiency and have been gradually applied in the fields of autonomous driving and robotics. For 3D-object detection, many scholars have proposed network structures, and these networks have superior performance [7–9]. Commonly used methods for 3D-object detection include converting point clouds into voxels [10] and pseudo images [11]. This also includes PointRCNN [12], which processes point clouds directly. Research progress has also been made in joint 3D-instance segmentation and object detection [13].

However, the general practice of the current 3D-object detection algorithm is to directly process the collected point cloud, and few scholars have considered removing the ground

first. The main reason is that the accuracy and efficiency of the current ground filtering algorithms have difficulty in meeting the application requirements of 3D-object detection. For example, the ground filtering accuracy is insufficient, which will result in part of the ground being retained and some objects being removed. This will seriously affect the object detection accuracy. At the same time, computational efficiency is an important indicator of 3D-object detection. If the ground filtering efficiency is insufficient, it is also impossible to be applied. Therefore, a fast and high-precision ground filtering method is currently needed.

Since the elevation changes of most scenes are relatively small, many researchers treat the ground surface as a flat surface [14]. Fan et al. builds a plane function by using RANSAC. The ground points whose distance to the optimal plane is within the threshold are recognized [15]. When the ground is uneven, this method has obvious defects. In order to improve the fitting accuracy in large scenes, Golovinskiy filters out the ground points locally [16].

As the whole ground cannot be a flat plane. Part of the ground is mostly a flat plane. However, they only filter the ground points through the plane fitting method and do not consider the problems that may be caused by the local plane fitting. Similarly, this paper uses a local method to filter out the ground points. The proposed method first divides the point cloud into several blocks, which is essential to obtain a local point cloud. This paper analyzes the problems that may be caused by local ground fitting, and proposes effective solutions.

For the efficiency and accuracy of RANSAC, Universal-RANSAC [17] conducted a detailed analysis and proposed a comprehensive solution. Similarly, we analyze the steps of RANSAC in detail and optimize the two key steps of sampling and determining the optimal plane, respectively.

- Sampling: RANSAC uses a completely random approach. The premise of this approach is that we have absolutely no idea what the data is like. However, in practical applications, prior knowledge of the data is known. GroupSAC [18] considers points within a class to be more similar, and points in a dataset are grouped according to some similarity. Sampling starts with the largest cluster as there should be a higher proportion of inliers here. In the process of LiDAR ground filtering, the heights of the two adjacent parts of the ground are essentially the same, and thus we can first estimate the height of the ground and set constraints. Our scheme can also deal with the problems caused by local ground fitting.
- Determining the optimal plane: After RANSAC calculates the model, the number of points that satisfy the parameters in all sets is calculated. Preemptive RANSAC [19] first generates multiple models, and then a selected subset is used to rank the generated models according to the objective function score. The first few are selected, and several rounds of similar sorting are performed to select the best model. We also adopt this idea. Multiple models are generated, and the best one is selected. This avoids multiple iterations and improves the efficiency.

The contributions of this article are as follows.

1. We propose an improved RANSAC. We analyze the principle of GroupSAC, design the sampling method and effectively solve the problems that may be caused by local point cloud filtering. We analyze Preemptive RANSAC and devise a method for determining the optimal plane. Based on these two key steps of RANSAC, a LiDAR ground filtering method is proposed.
2. We experimentally verify that the accuracy and efficiency of the proposed method are higher than the current commonly used methods. The filtering results obtained by the proposed method can better preserve the details in the point cloud.
3. We explore the application of point cloud filtering methods to 3D-object detection. The proposed method can improve the accuracy of 3D-object detection to a certain extent without affecting the efficiency.

2. Related Work

2.1. LiDAR Filtering

In recent years, researchers have proposed a variety of ground filtering algorithms, which can be classified according to different criteria. For example, filtering algorithms can be divided into urban pavement and wild vegetation according to the ground type. According to the processing method, these can be divided into single-step filtering and iterative filtering. After sorting out the ground filtering algorithms, the current mainstream algorithms are divided into three categories according to the point-cloud division and processing methods. These are the ground filtering algorithm based on morphology, the ground filtering algorithm based on space division and the ground filtering algorithm based on iterative least square interpolation.

Morphological filtering was the earliest filtering algorithm applied to LiDAR. The specific steps of this method are to divide the point-cloud data into grids, and the grid elevation information is used to erode the non-ground points to extract the ground points [20]. The progressive morphological filter gradually increases the window size, and according to the window size, the elevation difference threshold information is used to retain ground points and remove points from non-ground objects [21].

Pirotti used a multi-dimensional grid to apply a progressive morphological filter to remove non-ground points [22]. The algorithm does not require multiple iterations and can optimize the speed; however, it relies heavily on reflectance information. Trepekli evaluated the performance of morphological filter, and the results show that the performance of morphological filter on uniform surface is satisfactory [23].

This method generally requires interpolation and gridding before data processing, which will cause damage to the original terrain features. Furthermore, this kind of method only uses the lowest point of the window as the ground point. Assuming that the roof area is large and the ground is not included in the window, there will be errors in the filtering results, and this method is not applicable. In addition, the size of the structural window and the setting of the elevation threshold are the main factors that affect this filtering. This also leads to the impracticality of these methods.

Ground filtering based on space division is a mixture of grid-based filtering and three-dimensional voxel-based filtering. The grid-based filtering is to grid the horizontal plane of the point cloud space. Thrun et al. proposed a filtering algorithm based on the minimum–maximum height difference [24]. The ground filtering based on a two-dimensional grid uses local ground information instead of global continuity information for filtering. This method is susceptible to noise or external calibration of the sensor, and thus the performance is not stable.

Three-dimensional voxels are based on a plane grid and divide the three-dimensional space into several sets according to the elevation information of the point cloud [25]. This type of algorithm generally distinguishes ground voxels from non-ground voxels by judging the average height or variance value of the points within the voxels [26].

The filtering method based on iterative linear least squares interpolation was first proposed by Kraus et al. [27]. This method can obtain the terrain surface well; however, the obvious drawback of this method is that the filtering parameters need to be constantly adjusted to adapt to different types of terrain. Koebler proposed a layered robust linear interpolation method based on least squares [28]. This method is suitable for steep areas and forest areas. Qin proposed a region growth filtering based on moving-window weighted iterative least squares fitting [29]. This method can effectively remove buildings and vegetation; however, it still requires further improvement for the removal of bridges and objects at the edge.

Gao used least squares interpolation in the framework of road extraction to restore the elevation information of the blocked sections of the overpass [30]. This type of algorithm needs to satisfy two conditions. First, the lowest point of elevation value within a certain area must be a ground point. Second, the distribution of ground points conforms to the

quadratic surface distribution, and other points are higher than the surface. In short, the application of ground filtering methods based on least squares is limited.

2.2. RANSAC

The goal of classic RANSAC [31] is to continuously attempt different target space parameters to maximize the objective function. This is a random, data-driven process. The estimated model is generated by iteratively randomly selecting a subspace of the dataset. The estimated model is then leveraged and tested with the remaining points in the dataset to obtain a score. Finally, the estimated model with the highest score is returned as the model for the entire dataset. Classical RANSAC has three main limitations, namely efficiency, accuracy, and degradation. There are many improvements to these limitations of the classical approach.

Under the condition of prior knowledge, the minimum subset sampling method can effectively reduce the sampling times. The main idea of NAPSAC [32] is to regard the n-dimensional space of the dataset as a hypersphere, and as the radius decreases, the outliers decrease faster than the inliers. PROSAC [33] uses the result of matching the initial set of points as the basis for sorting, and thus that the samples that are most likely to obtain the best parameters will appear earlier, which improves the speed. Similar to NAPSAC, the classical algorithm begins to calculate the parameters after the sampling is completed, while some algorithms verify whether the sampling results are suitable for the parameter calculation after the sampling is completed.

The model calculation is to calculate the parameters according to the minimum set selected in the previous step to obtain the model. Prior knowledge is used for model validation, such as matching point sets with circles. When verifying, it is not necessary to verify all the points in the dataset but only to verify within a radius of the model.

The verification parameter is to calculate the number of points satisfying the parameter in all sets after obtaining the parameters generated by the minimum set. $T(d, d)$ test selects d points that are much smaller than the data set as the test. Only when these d points are all in-class points, are the remaining points are tested; otherwise, the current model is discarded. The Bail-Out test [34] selects several points in the set for testing. If the proportion of inliers is significantly lower than the proportion of inliers in the current best model, the model is discarded. The SPRT test [35,36] randomly selects a point and calculates the probability of conforming to the current model and the probability of not conforming. When the probability ratio exceeds a certain threshold, the current model is discarded.

The final converged RANSAC result may be affected by noise and is not the globally optimal result. This effect requires the addition of a post-processing of model refinement. When the current optimal result appears in the iterative process, Lo-RANSAC [37] re-samples from the inliers of the returned result to calculate the model by setting a fixed number of iterations and then selecting the optimal local result as the improved result. The idea of the error propagation method [38] is consistent with Lo-RANSAC, since the initial RANSAC results are generated from a noisy dataset, and thus this error propagates to the final model.

Universal-RANSAC [17] analyzes and compares various methods to optimize the key steps of RANSAC. The algorithm flow chart of Universal-RANSAC is shown in Figure 1. Its minimum sampling method adopts PROSAC, its model verification adopts SPRT test, and its detail optimization adopts Lo-RANSAC. In this paper, the key steps of RANSAC are optimized according to the characteristics of point clouds, and an efficient and robust LiDAR filtering method is proposed.

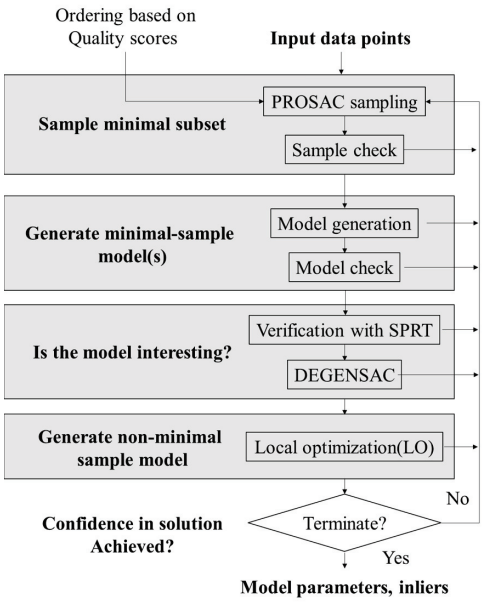


Figure 1. Flowchart of Universal-RANSAC.

2.3. 3D Object Detection

3D-object detection in urban environments is a challenge, requiring the real-time detection of moving objects, such as vehicles and pedestrians. In order to realize real-time detection in large-scale point clouds, researchers have proposed a variety of methods for different requirements.

Lang et al. proposed the encoder PointPillars to learn the point-cloud representation in pillars [11]. By operating the pillar, there is no need to manually adjust the combination of points in the vertical direction. Since all key operations can be represented as 2D convolutions, end-to-end 3D point cloud learning can be achieved using only 2D convolutions. The point cloud information can be effectively utilized by this method, and the calculation speed is fast.

Shi et al. proposed PointRCNN to generate ground-truth segmentation masks from point clouds in the scene based on bounding boxes [12]. A small number of high-quality bounding box preselection results are generated while segmenting the foreground points. Preselected results are optimized in standard coordinates to obtain the final inspection results.

Considering the generality of the model, Yang et al. proposed STD [39]. Spherical anchors are exploited to generate accurate predictions that retain sufficient contextual information. The normalized coordinates generated by PointPool make the model robust under geometric changes. The box prediction network eliminates the difference between localization accuracy and classification score, which can effectively improve the performance.

Liu et al. proposed LPD-Net (large-scale place description network) [40]. The network uses an adaptive local feature extraction method to obtain the local features of the point cloud. Second, the fusion of feature space and Cartesian space can further reveal the spatial distribution of local features and learn the structural information of the entire point cloud inductively.

Zhang et al. proposed PCAN to obtain local point features and generate an attention map [41]. The network uses ball queries of different radii to aggregate the textual feature information of points. This method can learn important point cloud features.

To overcome the limitation of the small size of point clouds in general networks, Paigwar et al. proposed Attentional PointNet [42] using the Attentional mechanism to

focus on objects of interest in large-scale and disorganized environments. However, the preprocessing step of this method makes it computationally expensive.

Voxel CNN is adopted for voxel feature learning and precise location generation to save subsequent computation and encode representative scene features [43]. Features are then extracted, and the aggregated features can be jointly used for subsequent confidence predictions. This method combines the advantages of voxel and Pointnet to learn more accurate point cloud features.

3. LiDAR Ground Filtering

In this section, we introduce the proposed method. The sampling part is introduced first. We first analyze the principle of GroupSAC and the possible problems caused by point cloud segmentation, and based on this, we propose a method to constrain the sampled points. Then, the calculation method of the plane equation and the method of counting the number of points in the plane are introduced. Finally, based on the analysis of Preemptive RANSAC, a method to determine the optimal plane is proposed.

The flow chart is shown in Figure 2. First, the point cloud is observed and divided into several parts evenly according to the length and width. We determine the constraints and select n sets of points. n plane equation models are built, and the point cloud is downsampled. Then, we count the number of points within the range of each plane model, and select the top m models with the largest number of point clouds. The point cloud before downsampling is used to count the number of points within the plane model again. At this time, the selected plane model with the largest number of points is the optimal model. This process is repeated to obtain the ground model of the entire point cloud.

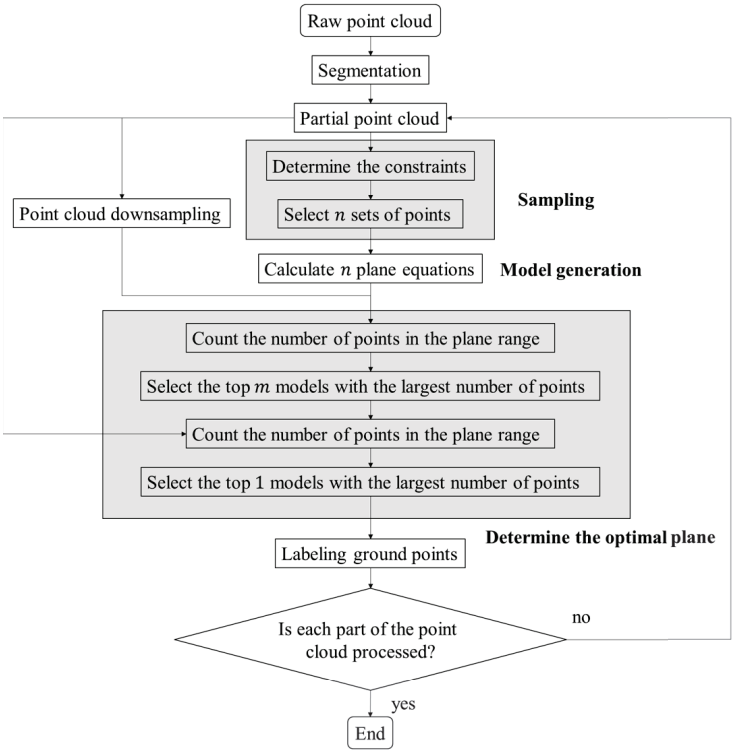


Figure 2. Flowchart of the proposed method.

3.1. Sampling

3.1.1. Principle Analysis of GroupSAC

In the original RANSAC and its many improved methods, the probability that point x is the target point in the random point sampling process obeys a Bernoulli distribution. That is, the possibility that a point is an inlier is considered to be independent of the other points. In the original RANSAC, the parameter estimation problem for an existing model from N data $\{x_j\}, j = 1 \cdots N$ is corrupted by interference. We suppose that the least number of data required for calculating the parameters of the model is m. For any minimal data set S with m data:

$$I_S \sim B(m, \epsilon) \tag{1}$$

where I_S is the number of all target points in S. $B(m, \epsilon)$ is the binomial distribution. "$\sim$" is the sign that I_S obeys the binomial distribution. ϵ is the parameter of the Bernoulli trial—that is, the target point possibility of S. Therefore, for the probability $P_{sum}(I_S = m)$ that all data in S are target points, the formula is:

$$P_{sum}(I_S = m) = \prod_{j=1|x_j \in S}^{m} P(I_j) = \epsilon^m \tag{2}$$

where I_j is a variable indicating that x_j is the target point. Despite the fact that many previous works consider that ϵ is not necessarily identical for various points. Furthermore, this inhomogeneous attribute is used to accelerate the sampling process. The target point probabilities of various points in these methods are still assumed to be independent of each other.

For many problems, there is a grouping between data. These grouped attributes tend to have high or low proportions of target points. Figure 3 is used as an example. We label the point cloud with different colors based on height. We can consider a group of similar colors as a point group. The green group is more likely to contain inliers than the blue group.

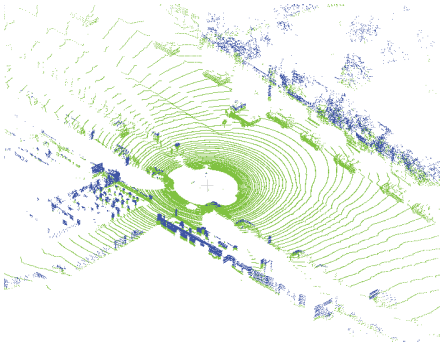


Figure 3. Schematic diagram of a point cloud grouping. According to the height, the point cloud is divided into two groups, the blue group and the green group. It is clear that the green group contains more inliers.

We hypothesize that the probability of inliers in these sets can be modeled by a two-class compound. They are the high inlier class and the low inlier class, respectively. The characteristic of the model is that the more data in the high inlier class, the lower the inlier ratio. The inlier ratio is about 0 in the low inlier class. The delta function is also called a generalized function. The larger the range of the function's definition domain, the smaller the range of the value domain. The value outside the domain is 0. The characteristics of the

delta function are highly compatible with the model. Thus, we use the delta function for modeling. The inlier ratio ϵ_i in any existing group G_i is:

$$\epsilon_i \sim \pi_h \delta(\epsilon_0) + \pi_z \delta(0) \tag{3}$$

where π_h and π_z are the mixture weights for the high inlier class and the zero inlier class. The inlier ratios of these two classes are ϵ_0 and 0, respectively. Therefore, the probability of having I_{G_i} inliers in G_i can be deduced as:

$$P(I_{G_i}) = \int_{\epsilon_i} P(I_{G_i}|\epsilon_i)P(\epsilon_i) = \pi_h P(I_{G_i}|\epsilon_i = \epsilon_0) + \pi_z P(I_{G_i}|\epsilon_i = 0) \tag{4}$$

That is to say, the distribution of inliers I_{G_i} for any existing group is:

$$I_{G_i} \sim \pi_h B(|G_i|, \epsilon_0) + \pi_z B(|G_i|, 0) = \pi_h B(|G_i|, \epsilon_0) \tag{5}$$

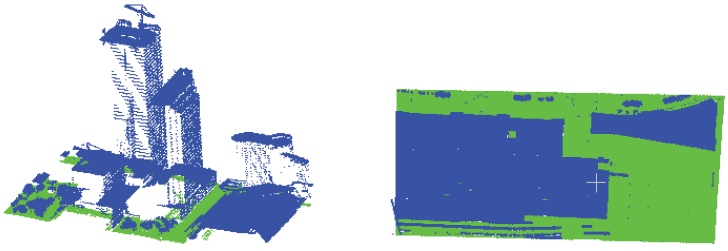
where $|G_i|$ is the number of points in G_i. Therefore, inliers are generated by only a part π_h of the groups, called the inlier groups [18]. In summary, we designed a method suitable for point clouds. Point clouds are grouped by height in order to find more suitable points.

3.1.2. Point Cloud Segmentation and Problem Description

We observe the horizontal and vertical slopes of the ground in the point-cloud data and make the ground of each part of the point cloud as plane as possible. The number of parts of the point cloud is as small as possible. Two problems may arise after the point cloud is divided into parts, and these are described as follows:

1. It is necessary to perform plane fitting processing for each part. Additional operations increase the calculation time.
2. When the building is tall, the number of points on the side of the building is more than the number of points on the ground as shown in Figure 4. The fitted plane is the side of the building, not the ground. When the plane fitting method is used for ground filtering, it will lead to incorrect results. When the area on the top of the house is large, this will also lead to wrong results.

Therefore, it is necessary to set constraints on the selection of random points.



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

Figure 4. Special cases. The green points are ground points, and the blue points are non-ground points. (**a**) The number of points on the side of the building is greater than the points on the ground. (**b**) The number of points on the top of the building is greater than the points on the ground.

3.1.3. Constraints of Sampled Points

In response to the above problems, this article proposes the following two constraints.

1. Two points are randomly selected from the three random points, the line of the two points is projected on the *xoz* and *yoz* planes, and the slope should be limited to $(-n, n)$. We use the coordinate values to calculate the slope of the projection of the line between the two points on the plane. For example, the coordinates of two

points are (x_1, y_1, z_1) and (x_2, y_2, z_2). The slope of the line connecting the two points projected on the xoz plane is $(z_2 - z_1)/(x_2 - x_1)$. The slope of the projection on the yoz plane is $(z_2 - z_1)/(y_2 - y_1)$. n refers to the slope threshold. According to the inclination of the ground in the point cloud, we set n manually. Assuming that the plane slope is not greater than 30 degrees, then $n = tan30^\circ \approx 0.577$.

2. We use the tools provided by the Point Cloud Library (PCL) [44] to observe the z coordinates (H_z, L_z) of the lowest and highest points on the ground of the point cloud that needs to be processed first. When randomly selecting the three initial coordinate points of the first point cloud part, the range of z-coordinates is limited to (H_z, L_z). Then, we calculate the optimal plane in the current point cloud under constraints. The range of z coordinates of the optimal plane is (h_z, l_z). When selecting the coordinates of three random points of the next point cloud, the z coordinate of these points are limited to $(h_z + t, l_z - t)$, where $t = h_z - l_z$. In short, the range of z coordinates of random points is determined according to the range of z coordinates of the optimal plane in the previous point cloud.

The increase of constraints will increase the calculation time; however, when selecting the points under constraints, the optimal plane can be obtained after a few iterations. Therefore, this method can reduce the number of iterations and improve the efficiency. This can solve the first problem mentioned above. By limiting the elevation and slope of the plane, it is easy to solve the second problem mentioned above.

3.2. Fitting Plane

According to the coordinates of three random points, the initial plane parameters are determined by the plane parameter calculation rules. The plane equation:

$$Ax + By - z + C = 0 \tag{6}$$

where A, B, C are parameters. The coordinates of the three points are $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$ and $P_3(x_3, y_3, z_3)$, respectively. We bring the coordinates of the three points into the equation to calculate the parameters:

$$A = \frac{(z_1 - z_3)(y_2 - y_3) - (z_2 - z_3)(y_1 - y_3)}{(x_1 - x_3)(y_2 - y_3) - (x_2 - x_3)(y_1 - y_3)} \tag{7}$$

$$B = \frac{(z_2 - z_3) - A(x_2 - x_3)}{y_2 - y_3} \tag{8}$$

$$C = z_1 - Ax_1 - By_1 \tag{9}$$

The plane equation can be obtained by substituting A, B, C into the Formula (1).

3.3. Counting the Number of Points on the Plane Range

For any point $P(x_p, y_p, z_p)$, the plane equation is $z = Ax + By + C$. We substitute (x_p, y_p) into the plane equation to obtain $z = Ax_p + By_p + C$. The distance from point P to the plane is $d = \mid z - z_p \mid$.

If the distance d from the point P to the fitting plane is less than the rejection threshold h_d, then the point P belongs to the plane. The rejection threshold h_d is manually set based on accuracy requirements of different scenarios.

3.4. Determining the Best Plane

The traditional process of determining the optimal plane is to first determine a plane by selecting random points. Then, the number of points in the plane is judged by the distance from the point to the plane, repeating this process until the plane with the largest number of points is obtained. There is no doubt that this process is inefficient.

The Preemptive RANSAC algorithm will evaluate a fixed number of hypothesis sets in parallel, multi-stage. The scoring mechanism selects candidate hypotheses from

a predefined number of candidate hypotheses with a small, fixed time to meet real-time requirements. This process replaces the scoring function of the classic RANSAC algorithm with a Preemptive scoring mechanism, thereby, avoiding over-scoring the useless candidate hypotheses distorted by noise. The scoring function $\rho(o,h)$ is used in the Preemptive RANSAC algorithm to represent the scalar value of the log-likelihood of the observed value. At this point, the log-likelihood function $L_i(h)$ of the candidate hypothesis with index h is as follows:

$$L_i(h) = \sum_{o=1}^{i} \rho(o,h) \tag{10}$$

where o is the observation, and there are N in total. h is the candidate hypothesis, $h = 1, \ldots, M$.

The Preemptive RANSAC algorithm defines the number of candidate hypotheses reserved by the function $f(i)$ for each stage as shown in the formula:

$$f(i) = \left\lfloor M 2^{-\left\lfloor \frac{i}{B} \right\rfloor} \right\rfloor \tag{11}$$

where $f(i)$ is modified after every B observations, $\lfloor \rfloor$ denotes downward truncation.

All observations are first randomly permuted, yielding a set of candidate hypotheses with indices $h = 1, \cdots, f(1)$. We compute the score $L_1(h) = \rho(1,h)$ for each candidate hypothesis, adjusting $i = 2$. Then, all candidate hypotheses are sorted according to the corresponding $L_{i-1}(h)$ values, and for $h = 1, \cdots, f(i)$, the first $f(i)$ candidate poses are selected to enter the next iteration. The iteration is stopped when $i > N$ or $f(i) = 1$. Otherwise, for the hypothesis $h = 1, \ldots, f(i)$, its score $L_i(h) = \rho(i,h) + L_{i-1}(h)$ is calculated, and the step of ranking the candidate hypotheses is continued.

Based on the analysis of Preemptive RANSAC, we designed a method to quickly determine the optimal plane. We set the constraints of initial point selection through the above method, and then selected n sets of points to calculate n plane equation models. The point cloud is downsampled to calculate the number of points within the bounds of each plane equation. We choose the m plane equations with the largest number of points. The above calculation is repeated in the selected plane equation using the origin point cloud. The plane equation with the largest number of points is the optimal plane equation. Among them, the parameters m and n have a great influence on the accuracy and speed of point cloud filtering. Therefore, comprehensive consideration should be given to the selection of parameters.

4. Experiments and Discussion

We use the tools provided by CloudCompare to label ground points and non-ground points in different colors. Then, we use the proposed method to label the ground points, and conduct a qualitative and quantitative comparative analysis. The point clouds used are from the KITTI data set [45] and the International Society for Photogrammetry and Remote Sensing (ISPRS) datasets [46].

- **KITTI**: The KITTI data set was jointly established by the Karlsruhe Institute of Technology (KIT) and Toyota Technological Institute at Chicago (TTI-C). It is currently the largest computer vision algorithm evaluation dataset in the world for autonomous driving scenarios. KITTI contains real data collected in urban, rural and highway scenes. A Velodyne HDL-64E 3D laser scanner was used to acquire point clouds. The laser scanner spins at 10 frames per second, capturing approximately 100 k points per cycle. The KITTI data is mainly ground scenes with many details, which can test the ability of the algorithm to process details.
- **ISPRS**: ISPRS provides two airborne data sets, including Toronto and Vaihingen. The data set is the data used for the test of digital aerial cameras performed by the German Association of Photogrammetry and Remote Sensing (DGPF). Toronto covers an area of approximately 1.45 km^2 in the downtown area. This area contains low-rise and high-

rise buildings. The average point density is 6 points/m^2. Vaihingen includes historic buildings with rather complex shapes and also trees. The average point density is 4 points/m^2. The terrain of Toronto is relatively flat, and the terrain of Vaihingen is uneven. The common feature is that the scene is complex. These two data can test the ability of the algorithm to handle complex scenes.

The parameter settings are as follows. The slopes of the point clouds used in this experiment are not very steep. Therefore, the point cloud is divided into 4 × 4 parts, and each part of the ground is close to the plane. The tools provided by PCL are used to observe the height of the ground of each part of the point cloud, and then we can set the height parameter of the fitted plane. The number of points in the point cloud used in the experiment is more than 100 k points, and the parameters m and n to determine the optimal plane are set to 100 and 10, respectively, at this time, the accuracy and efficiency can meet the requirements. If the number of points in the point cloud is small, the size of m and n can be appropriately increased to improve the accuracy.

4.1. Ground Filtering

We select two point clouds in the KITTI data set. Both point clouds are road scenes with 110 k points. We manually label the point cloud. As shown in Figure 5a, the green points are ground points, and the blue points are non-ground points. The red frame marked area in Figure 5a is enlarged as shown in Figure 5c.

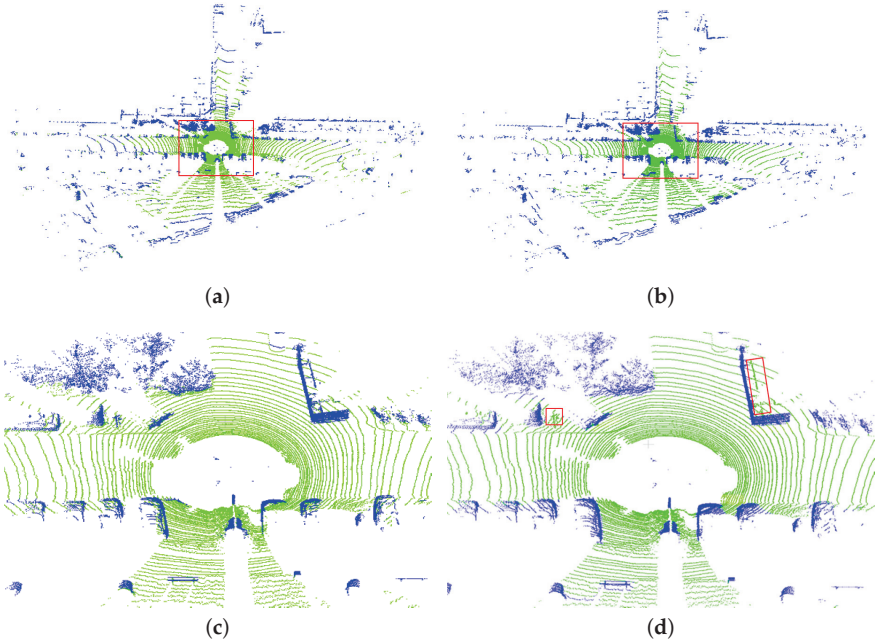


Figure 5. The result of the proposed method. The red box area in (**d**) is classified incorrectly. (**a**) The point cloud manually labeled. (**b**) The point cloud processed by the proposed method. (**c**) The red box area in (**a**). (**d**) The red box area in (**b**).

The height of the ground of the first point cloud part is about −3 to 0.2 m. We set the fitting plane height parameter (H_z, L_z) to $(-3, 0.2)$ and set the rejection threshold h_d to 1.5. After the parameter setting is completed, the point cloud is processed by the proposed method. The filtering result of the proposed method is shown in Figure 5b. The red frame marked area in Figure 5b is enlarged as shown in Figure 5d. Comparing Figure 5a,b, the method in this article can effectively distinguish ground points from other objects.

The results obtained by the proposed method are essentially consistent with the results of manual labeling. Comparing Figure 5c,d, the method in this article handles the details of the point cloud well, and can accurately distinguish ground points from non-ground points. A fact that can be demonstrated in Figure 6 as in Figure 5 is that the proposed method can better preserve details in the filtering results.

Toronto is an urban scene composed of 750 k points. The ground is relatively flat as shown in Figure 7a. We set the fitting plane height parameter (H_z, L_z) to (40, 50) and set the rejection threshold h_d to 5. The result is shown in Figure 7b. Comparing Figure 7a,b, it can be seen that the method in this article can better distinguish large buildings from the ground. At the same time, it can take into account the details of the ground, and some small objects can be distinguished from the ground.

Vaihingen is a village scene with a total of 720 k points. The ground in this village is uneven as shown in Figure 8a. We set the fitting plane height parameter (H_z, L_z) to (251, 270) and set the rejection threshold h_d to 15. The processing result of the proposed method is shown in Figure 8b. It can be seen from the figure that the method in this article can adapt to complex scene of the point cloud with uneven ground.

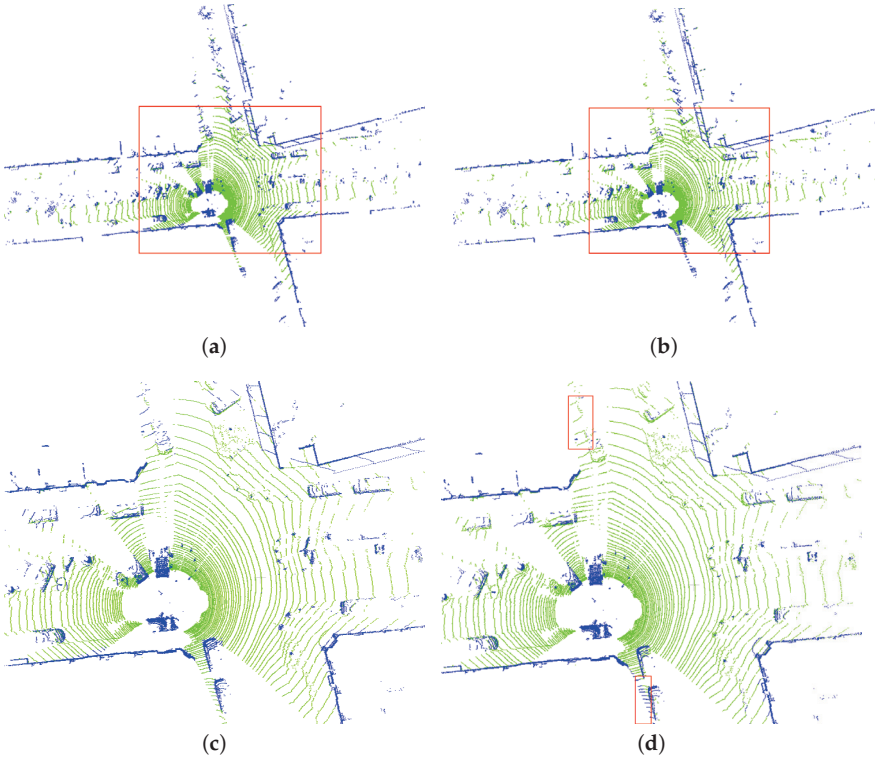
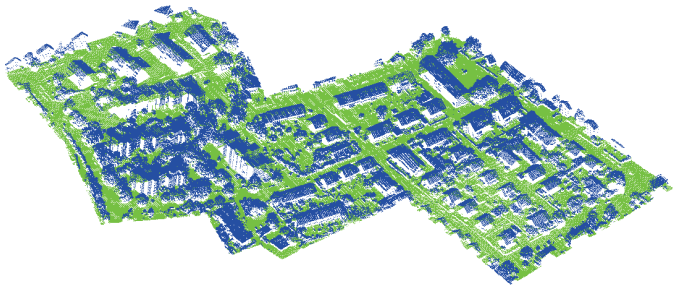


Figure 6. The result of the proposed method. The red box area in (**d**) is classified incorrectly. (**a**) The point cloud manually labeled. (**b**) The point cloud processed by the proposed method. (**c**) The red box area in (**a**). (**d**) The red box area in (**b**).

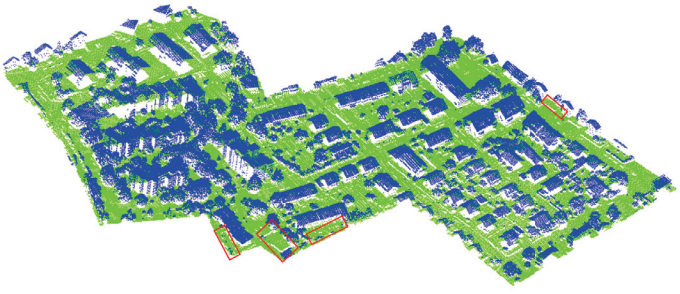
(**a**)

(**b**)

Figure 7. The result of the proposed method. (**a**) The point cloud manually labeled. (**b**) The point cloud processed by the proposed method. The red box area in (**b**) is classified incorrectly.



(**a**)



(**b**)

Figure 8. The result of the proposed method. (**a**) The point cloud manually labeled. (**b**) The point cloud processed by the proposed method. The red box area in (**b**) is classified incorrectly.

4.2. Accuracy Comparison

The proposed method is compared with eight methods. Among these methods are recent and classic. Method I is the original RANSAC. Method II is progressive TIN densification [47]. Method III is cloth simulation filter [48]. Method IV is the multiscale curvature classification [49]. Method V is active contours [50]. Method VI is regularization method [51]. Method VII is modified slope based filter [52]. Method VIII is hierarchical modified block-minimum [53]. The comparison of the first four methods is shown in Table 1, and we show the different performance of each method applied to different data. The comparison of the last four methods is shown in Table 2. We show the total error of filtering.

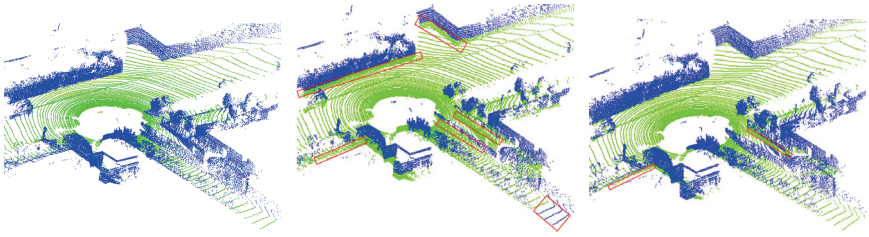
We use two data provided by ISPRS and two data in the KITTI dataset as experimental data. Error Type I, Error Type II, and the total error are used as evaluation indicators. The type I error represents the proportion of ground points erroneously assigned as nonground points, and the type II error represents the proportion of nonground points erroneously assigned as ground points. The total error is the proportion of all the point-cloud data that is misjudged and is used to evaluate the overall quality of the filtering results [54].

As shown in Table 1, compared with other methods, the error of the method in this article is relatively small. The mean value of the total errors of the proposed method is about 7.86%. The mean values of the total error of the remaining four methods are 18.4%, 9.57%, 8.5%, and 9.54%, respectively. Compared with Method I, the average error of the proposed method is reduced by about 10.54%. The comparison of the filtering results of these two methods is shown in Figure 9. Compared with the current commonly used methods, the average error of the proposed method is reduced by at least 0.64%. The proposed method has a better comprehensive performance on the KITTI dataset and the datasets provided by ISPRS.

Compared with other methods, the proposed method can adapt to complex scenes and deal with the details in the point cloud. The advantage of this method is that it has high filtering accuracy on relatively flat ground. The comparison with the last four methods is shown in Table 2. The average errors of other methods are significantly higher than those of the proposed method. This further confirms the high filtering accuracy of the proposed method.

Table 1. Comparison of the errors of the proposed method and other methods.

Method	Data	Type I Error (%)	Type II Error (%)	Total Error (%)
Proposed Method	Toronto	8.11	2.52	5.41
	Vaihingen	2.38	17.74	9.28
	KITTI1	0.94	10.15	6.25
	KITTI2	5.67	15.97	10.50
Method I	Toronto	10.59	25.62	18.45
	Vaihingen	24.65	19.75	22.53
	KITTI1	18.64	12.65	15.86
	KITTI2	16.47	17.57	16.75
Method II	Toronto	8.45	6.87	7.86
	Vaihingen	13.58	11.96	12.77
	KITTI1	3.42	12.21	7.96
	KITTI2	7.65	11.14	9.71
Method III	Toronto	14.56	4.78	9.27
	Vaihingen	10.64	6.98	8.40
	KITTI1	1.48	8.48	4.71
	KITTI2	14.68	8.57	11.67
Method IV	Toronto	12.54	6.86	8.74
	Vaihingen	5.76	17.86	11.46
	KITTI1	3.86	11.53	7.34
	KITTI2	13.75	8.64	10.65



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

Figure 9. The results of the comparison between the method proposed in this article and the RANSAC plane fitting method. Green points are the ground points, blue points are the non-ground points. The red boxes in (**b**,**c**) are classified incorrectly. (**a**) The point cloud manually labeled. (**b**) The point cloud processed by the RANSAC. (**c**) The point cloud processed by the method proposed in this article.

Table 2. Comparison of the total errors of the proposed method and other methods.

Data	Method V (%)	Method VI (%)	Method VII (%)	Method VIII (%)	Proposed (%)
Toronto	12.43	8.54	15.64	6.08	5.41
Vaihingen	9.06	11.53	14.53	11.46	9.28
KITTI1	7.75	9.68	16.34	5.75	6.25
KITTI2	14.64	15.57	11.91	16.45	10.50
average	10.97	11.33	14.61	9.94	7.86

4.3. 3D Object Detection Experiment and Efficiency Analysis

We explore the application of LiDAR ground filtering for 3D-object detection. We use the KITTI dataset. The vehicle is the detection object. We test three open-source 3D-object detection methods. Pretrained weights are used to detect objects in point clouds. The detection results of the unfiltered point cloud and the filtered point cloud are compared. The results are shown in Table 3. It can be clearly seen that when the filtered point cloud is used for 3D-object detection in simple or moderate situations, the detection accuracy is significantly improved.

In the process of object detection, ground points are often interference information. After removing the ground points, each object is in an isolated state, and the object detection algorithm only needs to match the detected object from multiple isolated objects. This can reduce the difficulty of object detection, thereby, improving the performance of object detection. However, when it is used for difficult 3D-object detection, the detection accuracy is slightly reduced. The main reason is that the filtering takes away a small part of the point cloud at the object. The original identification is more difficult, and it is more difficult to detect if some information is missing.

The LiDAR ground filtering experiment was conducted on a computer with Intel Core i7 3.19-GHz CPU and 16-GB RAM. The calculation time of the proposed method is about 20 ms to process a point cloud of 100 k points. Current 3D-object detection algorithms generally run on platforms with high computing power. Furthermore, better computing platforms have strong parallel computing capabilities, and thus the time used for ground filtering can be further reduced.

We randomly select 20 point clouds in the KITTI dataset, manually annotate the ground and non-ground points, and record the number of ground and non-ground points. We found that the ground points account for about 40–60% of the entire point cloud. The computation time of the 3D-object detection method is related to the number of points in the point cloud. The lower the number of points in the point cloud, the lower the runtime. Therefore, the filtered point cloud can improve the speed of 3D-object detection. The times for the three object detection methods are shown in Table 4. It can be clearly seen that the

detection time is significantly reduced. The results show that the proposed method does not have a large impact on the time of the 3D-object detection method.

Table 3. Comparison of accuracy before and after ground filtering.

		Raw Point Cloud (%)	No Ground Point Cloud (%)
CIA-SSD	Easy	89.59	90.57
	Mod	80.28	82.04
	Hard	72.87	74.46
CLOCs	Easy	89.16	90.34
	Mod	82.28	83.64
	Hard	77.23	75.85
SIENet	Easy	88.22	90.47
	Mod	81.71	85.15
	Hard	77.22	73.74

Table 4. Comparison of efficiency before and after ground filtering.

	Raw Point Cloud (ms)	No Ground Point Cloud (ms)
CIA-SSD	30	22
CLOCs	100	70
SIENet	80	55

5. Conclusions

In this paper, we proposed an improved RANSAC LiDAR ground filtering method. We evaluated the proposed method using point clouds with different characteristics and compared the filtering accuracy with a variety of commonly used methods. The results show that the filtering accuracy of this method was improved by about 10% compared with the original method and by about 1% compared with the current advanced method. Furthermore, this method has higher filtering efficiency.

The proposed method is intended to be applied to 3D-object detection. Ground filtering can improve object detection accuracy under simple and moderate conditions on the KITTI dataset. Furthermore, this can reduce the time of object detection. When the proposed method is applied to 3D-object detection methods, the influence of the filtering time on object detection can be ignored. This paper demonstrates that ground filtering can be used as an auxiliary method to improve the accuracy of 3D-object detection. Therefore, the LiDAR ground filtering method deserves further in-depth study.

Author Contributions: Conceptualization, J.L. and B.W.; methodology, B.W.; software, J.G.; validation, B.W. and J.G.; formal analysis, J.L.; investigation, B.W.; resources, J.G.; data curation, J.G.; writing—original draft preparation, B.W.; writing—review and editing, J.L.; visualization, J.G.; supervision, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the 13th Five-Year Plan Funding of China, the Funding 41419029102 and the Funding 41419020107.

Data Availability Statement: ISPRS datasets was provided by the German Society for Photogram metry, Remote Sensing and Geoinformation (DGPF). KITTI datasets was provided by Karlsruhe Institute of Technology (KIT) and Toyota Technological Institute at Chicago (TTI-C).

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

References

1. Niu, Z.; Xu, Z.; Sun, G. Design of a New Multispectral Waveform LiDAR Instrument to Monitor Vegetation. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1506.
2. Montealegre, A.L.; Lamelas, M.T.; Juan, D. A Comparison of Open-Source LiDAR Filtering Algorithms in a Mediterranean Forest Environment. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 4072–4085. [CrossRef]
3. Huang, S.Y.; Liu, L.M.; Dong, J. Review of ground filtering algorithms for vehicle LiDAR scans point-cloud data. *Opto-Electron. Eng.* **2020**, *47*, 190688-1–190688-12.
4. Zhao, H.; Xi, X.; Wang, C. Ground Surface Recognition at Voxel Scale From Mobile Laser Scanning Data in Urban Environment. *IEEE Geosci. Remote Sens. Lett.* **2019**, *99*, 1–5. [CrossRef]
5. You, H.; Li, S.; Xu, Y. Tree Extraction from Airborne Laser Scanning Data in Urban Areas. *Remote Sens.* **2021**, *13*, 3428. [CrossRef]
6. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [CrossRef] [PubMed]
7. Zheng, W.; Tang, W.; Chen, S. CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud. In Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI), Online, 2–9 February 2021.
8. Pang, S.; Morris, D.; Radha, H. CLOCs: Camera-LiDAR Object Candidates Fusion for 3D Object Detection. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020.
9. Li, Z.; Yao, Y.; Quan, Z.; Yang, W.; Xie, J. SIENet: Spatial Information Enhancement Network for 3D Object Detection from Point Cloud. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
10. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
11. Lang, A.H.; Vora, S.; Caesar, H. PointPillars: Fast Encoders for Object Detection From Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
12. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June2019.
13. Zhou, D.; Fang, J.; Song, X. Joint 3D Instance Segmentation and Object Detection for Autonomous Driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
14. Miadlicki, K.; Pajor, M.; Sakow, M. Real-time ground filtration method for a loader crane environment monitoring system using sparse LIDAR data. In Proceedings of the IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Gdynia, Poland, 3–5 July 2017.
15. Fan, W.; Wen, C.; Tian, Y. Rapid Localization and Extraction of Street Light Poles in Mobile LiDAR Point Clouds: A Supervoxel-Based Approach. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 292.
16. Golovinskiy, A.; Kim, V.G.; Funkhouser, T. Shape-based recognition of 3D point clouds in urban environments. In Proceedings of the IEEE International Conference on Computer Vision, San Francisco, CA, USA, 13–18 June 2010.
17. Raguram, R.; Chum, O.; Pollefeys, M. USAC: A Universal Framework for Random Sample Consensus. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2022–2038. [CrossRef]
18. Ni, K.; Jin, H.G.; Dellaert, F. GroupSAC: Efficient Consensus in the Presence of Groupings. In Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009.
19. Nister, D. Preemptive RANSAC for Live Structure and Motion Estimation. In Proceedings of the IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003.
20. Kilian, J.; Haala, N.; Englich, M. Capture and evaluation of airborne laser scanner data. *Int. Arch. Photogramm. Remote Sens.* **1996**, *31*, 383–388.
21. Zhang, K.; Chen, S.C.; Whitman, D. A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 872–882. [CrossRef]
22. Pirotti, F.; Guarnieri, A.; Vettore, A. Ground filtering and vegetation mapping using multi-return terrestrial laser scanning. *ISPRS J. Photogramm. Remote Sens.* **2013**, *76*, 56–63. [CrossRef]
23. Trepekli, K.; Friborg, T. Deriving Aerodynamic Roughness Length at Ultra-High Resolution in Agricultural Areas Using UAV-Borne LiDAR. *Remote Sens.* **2013**, *13*, 3538. [CrossRef]
24. Thrun, S.; Montemerlo, M.; Dahlkamp, H. Stanley: The Robot that Won the DARPA Grand Challenge. *J. Field Robot.* **2006**, *23*, 661–692. [CrossRef]
25. Zhao, G.; Yuan, J. Curb detection and tracking using 3D-LIDAR scanner. In Proceedings of the IEEE International Conference on Image Processing, Melbourne, Australia, 15–18 September 2013.
26. Douillard, B.; Underwood, J.; Kuntz, N. On the segmentation of 3D LIDAR point clouds. In Proceedings of the IEEE International Conference on Robotics & Automation, Shanghai, China, 9–13 May 2011.
27. Kraus, K.; Pfeifer, N. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS J. Photogramm. Remote Sens.* **1998**, *53*, 193–203. [CrossRef]
28. Kobler, A.; Pfeifer, N.; Ogrinc, P. Repetitive interpolation: A robust algorithm for DTM generation from Aerial Laser Scanner Data in forested terrain. *Remote Sens. Environ.* **2007**, *108*, 9–23. [CrossRef]
29. Qin, L.; Wu, W.; Tian, Y. LiDAR Filtering of Urban Areas with Region Growing Based on Moving-Window Weighted Iterative Least-Squares Fitting. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 841–845. [CrossRef]

30. Gao, L.; Shi, W.; Zhu, Y. Novel Framework for 3D Road Extraction Based on Airborne LiDAR and High-Resolution Remote Sensing Imagery. *Remote Sens.* **2021**, *13*, 4766. [CrossRef]
31. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
32. Myatt, D.R.; Torr, P.H.; Nasuto, S.J. NAPSAC: High Noise, High Dimensional Robust Estimation. In Proceedings of the British Machine Vision Conference, Cardiff, UK, 2–5 September 2002.
33. Chum, O.; Matas, J. Matching with PROSAC—Progressive Sample Consensus. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June 2005.
34. Capel, D. An Effective Bail-Out Test for RANSAC Consensus Scoring. In Proceedings of the British Machine Vision Conference, Oxford, UK, 5–8 September 2005.
35. Chum, O.; Matas, J. Optimal Randomized RANSAC. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1472–1482. [CrossRef] [PubMed]
36. Matas, J.; Chum, O. Randomized RANSAC with Sequential Probability Ratio Test. In Proceedings of the IEEE International Conference on Computer Vision, Beijing, China, 17–21 October 2005.
37. Chum, O.; Matas, J.; Kittler, J. Locally Optimized RANSAC. In Proceedings of the DAGM-Symposium Pattern Recognition, Magdeburg, Germany, 10–12 September 2003.
38. Raguram, R.; Frahm, J.; Pollefeys, M. Exploiting Uncertainty in Random Sample Consensus. In Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009.
39. Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; Jia, J. STD: Sparse-to-dense 3D object detector for point cloud. In Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2019.
40. Liu, Z.; Zhou, S.; Suo, C. LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis. In Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2019.
41. Zhang, W.; Xiao, C. PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
42. Paigwar, A.; Erkent, O.; Wolf, C. Attentional PointNet for 3D-object detection in point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
43. Shi, S.; Guo, C.; Jiang, L. PV-RCNN: Point-voxel feature set abstraction for 3D-object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
44. Aiswarya, G.; Valsaraj, N.; Vaishak, M. Content-based 3D image retrieval using point cloud library a novel approach for the retrieval of 3D images. In Proceedings of the International Conference on Communication and Signal Processing, Melmaruvathur, India, 6–8 April 2017.
45. Sithole, G.; Vosselman, G. Comparison of filtering algorithms. *Int. Arch. Photogramm. Remote Sens.* **2003**, *34*, 1–8.
46. Wang, B.; Frémont, V.; Rodríguez, S.A. Color-based road detection and its evaluation on the KITTI road benchmark. In Proceedings of the IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014.
47. Zhang, J.; Lin, X. Filtering airborne LiDAR data by embedding smoothness-constrained segmentation in progressive TIN densification. *ISPRS J. Photogramm. Remote Sens.* **2013**, *81*, 44–59. [CrossRef]
48. Zhang, W.; Qi, J.; Wan, P. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sens.* **2016**, *8*, 501. [CrossRef]
49. Evans, J.S.; THudak, A. A multiscale curvature algorithm for classifying discrete return LiDAR in forested environments. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 1029–1038. [CrossRef]
50. Elmqvist, M.; Jungert, E.; Lantz, F.; Persson, A.; Soderman, U. Terrain modelling and analysis using laser scanner data. *Int. Arch. Photogramm. Remote Sens.* **2001**, *34*, 219–227.
51. Sohn, G.; Dowman, I. Terrain Surface Reconstruction by the Use Of Tetrahedron Model With the MDL Criterion. *Int. Arch. Photogramm. Remote Sens.* **2002**, *34*, 336–344.
52. Roggero, M. Airborne Laser Scanning: Clustering in raw data. *Int. Arch. Photogramm. Remote Sens.* **2001**, *34*, 227–232.
53. Wack, R.; Wimmer, A. Digital Terrain Models From Airborne Laser Scanner Data—A Grid Based Approach. *Int. Arch. Photogramm. Remote Sens.* **2002**, *34*, 293–296.
54. Sithole, G.; Vosselman, G. Report: ISPRS Comparison of Filters. Available online: http://www.itc.nl/isprswgIII-3/filtertest/ (accessed on 27 December 2016).

remote sensing

MDPI

Article

Orbital Maneuver Optimization of Earth Observation Satellites Using an Adaptive Differential Evolution Algorithm

Qizhang Luo, Wuxuan Peng, Guohua Wu and Yougang Xiao *

School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China; qz_luo@csu.edu.cn (Q.L.); pengwuxuan@csu.edu.cn (W.P.); guohuawu@csu.edu.cn (G.W.)
* Correspondence: csuxyg@csu.edu.cn

Abstract: Earth observation satellite (EOS) systems often encounter emergency observation tasks oriented to sudden disasters (e.g., earthquake, tsunami, and mud-rock flow). However, EOS systems may not be able to provide feasible coverage time windows for emergencies, which requires that an appropriately selected satellite transfers its orbit for better observation. In this context, we investigate the orbit maneuver optimization problem. First, by analyzing the orbit coverage and dynamics, we construct three models for describing the orbit maneuver optimization problem. These models, respectively, consider the response time, ground resolution, and fuel consumption as optimization objectives to satisfy diverse user requirements. Second, we employ an adaptive differential evolution (DE) integrating ant colony optimization (ACO) to solve the optimization models, which is named ACODE. In ACODE, key components (i.e., genetic operations and control parameters) of DE are formed into a directed acyclic graph and an ACO is appropriately embedded into an algorithm framework to find reasonable combinations of the components from the graph. Third, we conduct extensive experimental studies to show the superiority of ACODE. Compared with three existing algorithms (i.e., EPSDE, CSO, and SLPSO), ACODE can achieve the best performances in terms of response time, ground resolution, and fuel consumption, respectively.

Keywords: orbit maneuver; orbit coverage analysis; earth observation satellite (EOS); differential evolution algorithm; ant colony optimization

Citation: Luo, Q.; Peng, W.; Wu, G.; Xiao, Y. Orbital Maneuver Optimization of Earth Observation Satellites Using an Adaptive Differential Evolution Algorithm. *Remote Sens.* **2022**, *14*, 1966. https://doi.org/10.3390/rs14091966

Academic Editors: Yue Wu, Kai Qin, Qiguang Miao and Maoguo Gong

Received: 28 March 2022
Accepted: 16 April 2022
Published: 19 April 2022

1. Introduction

Earth observation satellite (EOS) systems can acquire images of the Earth's surface via their remote sensing instruments. Due to the advantages such as large-scale observation coverage and high observation frequency, EOSs have been widely implemented to monitor and observe disasters such as earthquakes, floods, landslides, and debris flow [1–3]. Although the number of EOSs is continuously increasing, there are still several limitations to satisfy all kinds of user requirements. For example, when an earthquake occurs, EOSs are required to take ground images urgently to provide timely support for rescue operations. However, EOSs in their regular orbits may not be able to observe the earthquake area timely or clearly. Thus, an appropriately selected satellite needs to be transferred to a new orbit to provide better coverage properties, which is termed the orbit maneuver optimization problem.

Generally, the orbit maneuver optimization problem can be treated as a kind of orbit design problem [4–6]. Numerous studies have been carried out to investigate the orbit design problem. For example, Graham et al. [7] studied a minimum-time Earth-orbit transfers optimization problem using low-thrust propulsion with eclipsing. They developed an initial guess generation method to construct a useful guess and analyzed the approximate place where the spacecraft enters and exits the Earth's shadow. A similar problem was addressed by Wang et al. [8], who adopted a convex optimization method. Zhang et al. [9] investigated a minimum-fuel optimization problem using low-thrust in the circular restricted three-body scenario. By considering actuation uncertainties,

Mohammadi et al. [10] proposed a robust optimization approach for the impulsive orbit transfers optimization problem. In their study, the genetic algorithm, Monte-Carlo sampling, and surrogate model are combined to balance the optimization accuracy and time. Cheng et al. [11] developed a real-time optimal control approach based on multiscale deep neural networks for the orbit transfer problem of the solar sail spacecraft. In a recent study, Morante et al. [12] proposed a multi-objective optimization approach for an orbit-raising optimization problem, in which chemical, electrical, and hybrid trajectories are considered.

However, most of the orbit design problems aim to find an optimal orbit for improving orbit performance (e.g., coverage time and fuel consumption) [4,5,13]. Those studies assume that the satellite flies along a fixed orbit without orbit maneuvers and consider orbit elements as decision variables. For the cases in which orbit maneuvers are considered, there are few existing studies that mainly focus on reconfiguration problems of satellite constellations [14–16]. For example, McGrath et al. [17] presented a satellite constellation reconfiguration problem, in which a restricted low-thrust Lambert rendezvous scenario was included. Soleymani et al. [18] investigated an optimal mission planning problem of the reconfiguration process of satellite constellations. They applied a combination of particle swarm optimization and genetic algorithm to find the optimal departure and arrival positions of each satellite. He et al. [19] developed a physical programming method together with a genetic algorithm, to solve a multi-objective satellite constellation reconfiguration problem for disaster monitoring purposes. Wang et al. [20] proposed a hybrid-resampling particle swarm optimization method for an agile satellite constellation design problem, in which different types of sensors, the attitude maneuver of sensors, and different coverage performance indices are considered. To satisfy the requirements of emergency observation, a recent study proposed by Hu et al. [21] carried out a multi-objective optimization framework for the satellite constellation optimization problem.

It can be concluded that although many relevant studies have been published, the orbit maneuver optimization problem that optimizes maneuvers of a satellite is still a minor branch of orbit design problems and is rarely investigated. Hence, in this study, we make effort to address the orbit maneuver optimization problem from a scheduling perspective. Specifically, since a satellite can transfer its orbit by conducting an impulsive maneuver at a specific time instance and the maneuver result would affect the orbit performance, it would be crucial to determine the reasonable magnitude and direction of the impulse, as well as the maneuver moment. Different from most of the previous studies that aim to determine the promising position (i.e., orbit elements) of a satellite, our study optimizes the orbit maneuver in terms of velocity increments for an impulsive maneuver and the maneuver moment. Meanwhile, our study considers multiple satellites and the most suitable satellite would be selected to execute the task according to scheduling results.

On the other hand, to improve the service quality, diverse user requirements are being considered in the orbit maneuver optimization in recent years. For instance, since the fuel capacity is limited and the remaining fuel affects the lifetime of a satellite, some users may require a low fuel consumption solution. In case of some emergency tasks that need to be accomplished at all costs, the users may want the satellite to respond to observation requests as quickly as possible. Further, in some rescue operations, the orbit altitude is the optimization objective since an appropriate orbit altitude that can provide higher ground resolution is crucial. Therefore, we build three models that, respectively, optimize three objectives, including response time, ground resolution, and fuel consumption to satisfy diverse user requirements. Meanwhile, since we focus on EOS, specific constraints such as the resolution constraint are included in models.

Since the studied problem considers orbit maneuvers at every second as decision variables, the search space would be very large. Meanwhile, specific constraints of EOS would increase the difficulties of solving the problem. All of the above reasons propose challenges for solving the problem. In this regard, evolutionary algorithms would be a promising solution method owing to their powerful and effective search capabilities. Previously, evolutionary algorithms have been widely employed to address the orbit design

problem. The algorithms used mainly include particle swarm optimization [20,22,23], genetic algorithms [16,21,24], and hybrid algorithms [5,25]. For example, Shirazi [25] applied a hybridization of the genetic algorithm and simulated annealing to a multi-objective orbit maneuver optimization problem. Based on the particle swarm optimization (PSO) algorithm, Pontani et al. [22] solved four kinds of impulsive orbital transfer problems, focusing on the optimization of impulsive transfers between two coplanar and non-coplanar, circular and elliptic orbits, respectively. Yao et al. [26] investigated the application of an improved DE algorithm on an orbit design problem by adding self-adaption and stochastic mechanisms. To optimize coverage-related metrics, as well as the number and semi-major axes of satellites in multiple constellations, Hitomi et al. [27] proposed a variable-length chromosome-based evolutionary algorithm.

Particularly, our studied problem can be treated as a continuous optimization problem. As a simple and efficient evolutionary algorithm, especially for continuous optimization, differential evolution (DE) which has rarely been implemented by previous related studies would be a promising candidate for addressing our problem. However, due to the well-known no-free-lunch theorem [28], the same optimization algorithm with the same configurations may have different performances on different problems. We have three models with different constraints and different optimization objectives, which propose challenges for optimizers. Moreover, DE highly depends on the configuration of genetic strategies and control parameters [29]. It would be time-consuming to find effective combinations of configurations to obtain high-quality solutions on different optimization models by using the same algorithm. Previously, many techniques have been developed to relieve this issue, such as ensemble and adaption techniques [28,30,31]. In this study, we implement the adaption technique to DE. Specifically, we form the genetic strategies and parameters of DE into a directed acyclic graph, in which each path indicates a combination of the genetic strategies and parameters. As the pheromone trails and property always enable the ant colony to find a reasonable path from the graph, an ant colony optimization (ACO) is adopted to search for effective combinations during the evolution. The hybridization of ACO and DE exhibits the effective search capability of DE that has been proved in previous studies [4,26,32]. Furthermore, it can dynamically optimize the algorithm configurations to improve the adaptive capability of DE, such that higher-quality solutions can be obtained for all three optimization models.

In summary, this paper has the following contributions.

(i) We investigate the orbit maneuver optimization problem considering diverse user requirements. In the problem, a satellite is selected from a set of satellites and transferred to a new orbit based on appropriate maneuvers (i.e., the velocity increment and maneuver moment) to respond to an emergency observation request. By analyzing orbit coverage and dynamics, we build three optimization models that optimize response time, ground resolution, and fuel consumption, respectively, to satisfy different user requirements.

(ii) To solve the proposed optimization models, we implement an adaptive differential evolution based on graph search. In the algorithm, key algorithm components (i.e., crossover strategies, mutation strategies, and control parameters) are formed into a directed acyclic graph and an ACO is adopted to find reasonable combinations of configurations during the evolution. The implemented algorithm is a hybrid of ACO and DE, therefore it is named ACODE.

(iii) We conduct simulation experiments to verify the efficiency of ACODE. The ACODE is compared with three representative algorithms including EPSDE [33], CSO [34], and SLPSO [35] in simulation scenarios where multiple EOSs are requested to observe a ground target. The simulation results show the superiority of ACODE.

This paper is organized as follows. Section 2 details the orbit coverage and dynamics analysis, as well as three optimization models. Sections 3 and 4 introduce the solution method and simulation experiments, respectively. Finally, the conclusions are remarked by Section 5.

2. Problem Description

In this section, we elaborate on the orbit maneuver optimization problem based on orbit coverage and dynamics calculations, followed by three optimization models with different optimization objectives (i.e., response time, fuel consumption, and ground resolution). As a part of the satellite system design, orbit maneuver optimization is associated with many complicated environmental factors. Therefore, some reasonable assumptions are adopted to simplify the problem.

(i) There are some perturbations (e.g., atmospheric drag, solar radiation pressure, and third body effects) that have negative impacts on the operation of the satellite. We only consider J_2 perturbation of Earth oblateness in the model, which is a common assumption in existing studies on orbit design problems [6,32,36].

(ii) Assume that the sensor equipped on each satellite is visible to the ground target when the satellite flies in the sunshine and the sunshine time is from 6:00 to 18:00 local time. Further, the other factors that may affect the imaging such as clouds and weather conditions, as well as the altitude of ground targets are assumed to be negligible.

(iii) Each satellite is assumed to be independent. Therefore, the orbit maneuver of a satellite does not affect the flying of another satellite.

(iv) The time required by the satellite to process task information and start the rocket engine is assumed to be negligible.

(v) The ground target is assumed to be a point target. Hence, the ground target can be imaged by the satellite once the satellite passes over it.

Main notations used in this section are displayed in Table 1.

Table 1. Notations.

Notations	Description
λ, λ_h, λ_{max}, and λ_{min}	Actual, horizon-, maximum, and minimum Earth's angular radius
η, η_c, η_h, η_{max}, and η_{min}	Actual, center, horizon-, maximum, and minimum boresight angle of the sensor
R_E	Earth's radius
r_{sat}	Distance between the Earth's center and the satellite
γ	Intermediate angle
$[lat_s, lon_s]$	Latitude and longitude of a subsatellite point
a	Semimajor axis
e	Eccentricity
i	Inclination
Ω and $\dot{\Omega}$	Longitude of ascending node and its time variation
ω and $\dot{\omega}$	Argument of perigee and its time variation
θ, M, and E	True, mean, and eccentric anomaly
P	Period for an orbit
μ	Earth's gravitational parameter
t_0	Time since perigee at the initial epoch
h	Angular momentum of the satellite
$\mathbf{r}^O$ and $\mathbf{r}^I$	Position vectors of the satellite in PQW and ECI frames
$\mathbf{v}^O$	Velocity vector of the satellite in PQW frame
D_{imag}	Satellite altitude over the ground target
H_{new}	Orbital altitude of the satellite after maneuvering
Δv and Δv_{max}	Velocity increment and the allowed maximum velocity increment for maneuvering
t_m	Maneuver moment
t_t	Time when the satellite receives the observation task
T_r	Response time
$[t_s, t_e]$	Sunshine time window of a ground target
T	Maximum response time required by users
R	Minimum ground resolution required by users

2.1. Orbit Coverage Analysis

The visibility between a satellite and a ground target depends on many factors, such as the location of the ground target (i.e., longitude and latitude), the orbit elements, and the field of view (FOV) of the satellite. To conduct the orbit coverage analysis, we assume that the Earth is a round body, the orbit is approximately circular, and the FOV on the ground is rectangular as in [32,37]. The ground target is visible to the satellite when it lies in the FOV, which can be determined by calculating the longitudes and latitudes of four vertices. A typical satellite coverage on the Earth is shown in Figure 1.

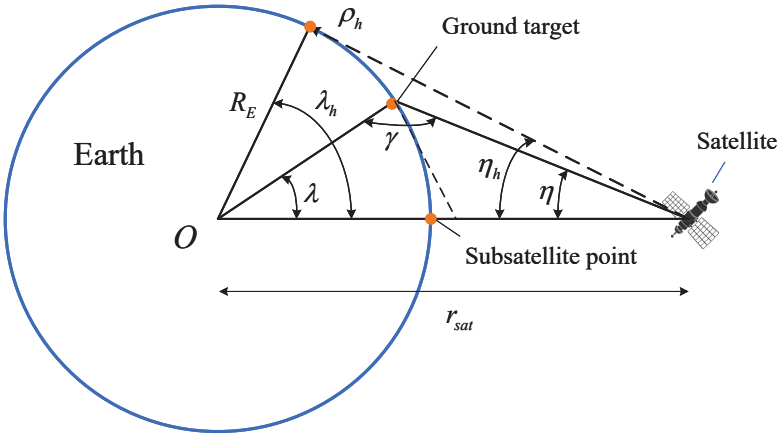


Figure 1. Satellite coverage on the Earth.

As Figure 1 shows, the Earth's angular radius λ_h defines the half-ground range that may visible to the satellite, which can be expressed by

$$\cos \lambda_h = \frac{R_E}{r_{sat}}, \tag{1}$$

where R_E is the Earth's radius, and r_{sat} is the distance between the Earth's center and the satellite. The slant range to the horizon, ρ_h, can be written as

$$\rho_h = \sqrt{{r_{sat}}^2 - {R_E}^2}. \tag{2}$$

However, in practical applications, due to some limitations such as the imaging angle of the sensor and sunshine conditions, the actual half ground range would be smaller than λ_h. Hence, a general expression for the slant range to any point, ρ, can be expressed by [38]

$$\rho = R_E \cos \gamma + r_{sat} \cos \eta, \tag{3}$$

$$\sin \gamma = \frac{r_{sat} \sin \eta}{R_E}, \tag{4}$$

where γ is the intermediate angle and η is the boresight angle of the satellite (i.e., half of the sensor angle). Afterward, the half-ground range from the subsatellite point can be calculated by

$$\sin \lambda = \frac{\rho \sin \eta}{R_E}. \tag{5}$$

For the satellite equipped with a scanning sensor, the geometry of the FOV is no longer symmetrical about the subsatellite point, requiring more processing to obtain the ground range angle. Given the center boresight angle η_c of the satellite, the maximum and

minimum ground-range angles from the subsatellite point can be obtained. Specifically, the maximum and minimum boresight angles can be written as [38]

$$\eta_{max} = \eta_c + \eta, \tag{6}$$

$$\eta_{min} = \eta_c - \eta. \tag{7}$$

Then, the maximum and minimum Earth's angular radiuses (i.e., λ_{max} and λ_{min}) can be obtained according to Equations (5)–(7). Since the sensor of the satellite can be rotated on multiple axes, in this study we assume that the sensor half-angle equals η_{max} and the Earth's angular radius equals λ_{max} for convenience. Define the latitude and longitude of the subsatellite point as $[lat_s, lon_s]$, the latitudes and longitudes of the four vertices of the FOV can be calculated by $[lat_s + \lambda_{max}, lon_s + \lambda_{max}]$, $[lat_s + \lambda_{max}, lon_s - \lambda_{max}]$, $[lat_s - \lambda_{max}, lon_s + \lambda_{max}]$, and $[lat_s - \lambda_{max}, lon_s - \lambda_{max}]$, respectively.

According to the latitude and longitude information of the FOV, the latitude and longitude information of the ground target, the positions of the satellite at each moment, as well as the right ascension of Greenwich at the initial moment, we can obtain the key orbit performance indices of a satellite [39], such as the response time [32,40]. The response time is defined as the time required from when a request is received to observe a ground target until the satellite can observe it. The method that assesses whether the target lies in the FOV at moment t, as well as the response time t_{imag} can be found in [32]. Moreover, the calculation method of the latitude and longitude of the subsatellite point at each moment is introduced in the next section.

2.2. Orbit Dynamics Model

The position of a satellite in its orbit can be obtained by using six orbit elements, including semimajor axis a, eccentricity e, inclination i, longitude of ascending node Ω, argument of perigee ω, and true anomaly θ, as Figure 2 shows. By using the orbit elements, we can calculate the position of the satellite, as well as the latitude and longitude of each subsatellite point at each moment. In this section, we briefly introduce the calculation methods, and more detailed derivation steps can be found in [41].

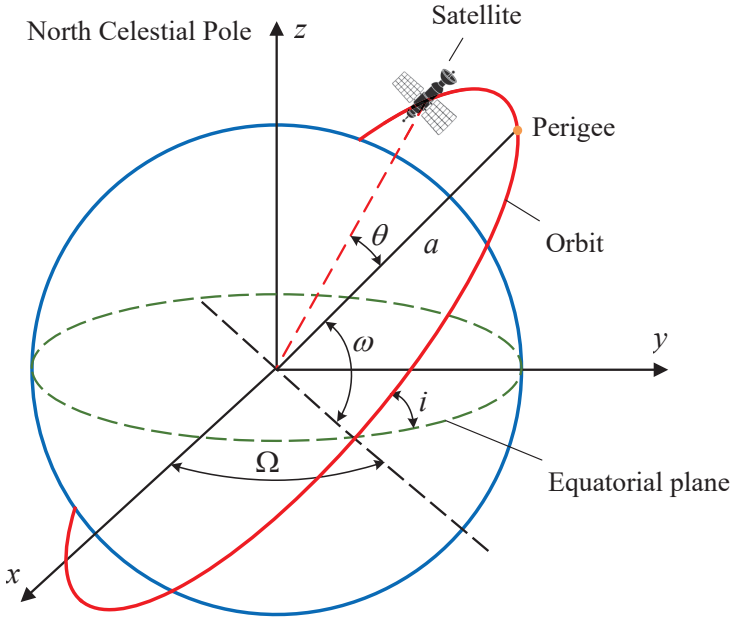


Figure 2. Geocentric equatorial frame and the orbital elements.

Given a satellite flying around the Earth, it is well-known that the period P for an orbit of the satellite is calculated by

$$P = \frac{2\pi}{\sqrt{\mu}} a^{3/2}, \tag{8}$$

where μ is the Earth's gravitational parameter. Then, the time t_0 since perigee at the initial epoch can be calculated as

$$t_0 = \frac{M}{2\pi} P, \tag{9}$$

where M is the initial mean anomaly. According to Kepler's equation, M can be calculated by

$$M = E - e \sin E, \tag{10}$$

where E is the eccentric anomaly, which yields the relation with true anomaly θ as

$$\tan \frac{E}{2} = \sqrt{\frac{1-e}{1+e}} \tan \frac{\theta}{2}. \tag{11}$$

Given a time change Δt, the longitude of ascending node Ω, argument of perigee ω at the moment $t = t_0 + \Delta t$ can be expressed by

$$\Omega = \Omega + \dot{\Omega} \Delta t, \tag{12}$$

$$\omega = \omega + \dot{\omega} \Delta t, \tag{13}$$

where $\dot{\Omega}$ and $\dot{\omega}$ are time variations of Ω and ω, which are determined by J_2 perturbation of Earth oblateness. The expressions of $\dot{\Omega}$ and $\dot{\omega}$ are written as

$$\dot{\Omega} = \left[\frac{3}{2} \frac{\sqrt{\mu} J_2 R_E^2}{(1-e^2)^2 a^{7/2}} \right] \cos i, \tag{14}$$

$$\dot{\omega} = \dot{\Omega} \frac{5/2 \sin^2 i - 2}{\cos i}, \tag{15}$$

where $J_2 = 1.083 \times 10^{-3}$. The orbit elements are updated by repeating Equations (9)–(13) at each moment t. Meanwhile, the newly found true anomaly θ at the moment t can be used to calculate the state vector of the satellite in the perifocal coordinate coordinate system (PQW). The satellite position vector $\mathbf{r}^O$ and velocity vector $\mathbf{v}^O$ in the PQW frame can be expressed by

$$\mathbf{r}^O = \frac{h^2}{\mu} \frac{1}{1 + e \cos \theta} \begin{Bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{Bmatrix}, \tag{16}$$

$$\mathbf{v}^O = \frac{\mu}{h} \begin{Bmatrix} -\sin \theta \\ e + \cos \theta \\ 0 \end{Bmatrix}, \tag{17}$$

where h is the angular momentum of the satellite, yielding a relation with the semimajor axis a as below

$$a = \frac{h^2}{\mu} \frac{1}{1-e^2}. \tag{18}$$

Particularly, the position vectors $\mathbf{r}^O$ can be transformed to the Earth-centered inertial (ECI) frame through the transformation matrix $R^{I/O}$ ($C \equiv cos$ and $S \equiv sin$) written as

$$R^{I/O} = \begin{bmatrix} C_\omega C_\Omega - C_i S_\omega S_\Omega & -S_\omega C_\Omega - C_i S_\Omega C_\omega & S_i S_\Omega \\ C_\omega S_\Omega + C_i S_\omega C_\Omega & -S_\omega S_\Omega + C_i C_\Omega C_\omega & -S_i C_\Omega \\ S_\omega S_i & C_\omega S_i & C_i \end{bmatrix}, \tag{19}$$

by

$$\mathbf{r}^I = R^{I/O}\mathbf{r}^O, \tag{20}$$

where $\mathbf{r}^I$ is the satellite position in the ECI frame. Meanwhile, $\mathbf{r}^I$ can be expressed in the Rotating Earth-fixed frame by [4]

$$\mathbf{r}^{I'} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{r}^I, \tag{21}$$

which can be written in vector notation

$$\mathbf{r}^{I'} = X\hat{\mathbf{I}} + Y\hat{\mathbf{J}} + Z\hat{\mathbf{K}}. \tag{22}$$

Define a notation $r = \sqrt{X^2 + Y^2 + Z^2}$, the latitude and longitude of the subsatellite point can be calculated by

$$lat_s = \sin^{-1}(Z/r), \tag{23}$$

$$lon_s = \begin{cases} \cos^{-1}\left(\frac{X}{r\cos lat_s}\right), & \frac{Y}{r} > 0 \\ 360^\circ - \cos^{-1}\left(\frac{X}{r\cos lat_s}\right), & \text{otherwise} \end{cases}. \tag{24}$$

Based on the above equations, the position of the satellite, as well as the latitude and longitude of each subsatellite point at each moment can be obtained.

Furthermore, in this study, the orbit maneuver focuses on how to move a satellite in the same plane, which can be treated as a co-orbital rendezvous problem [42]. In the co-orbital rendezvous problem, two satellites are assumed to be located in the same orbit and one satellite maneuvers its orbit by two-impulse Hohmann transfer to catch up with the other one. Therefore, a velocity increment Δv at the moment t_m is considered to calculate the orbit elements before and after maneuvering based on orbit equations.

2.3. Formulation of the Optimization Problem

As mentioned above, the optimization problem aims to find appropriate velocity increment Δv and maneuver moment t_m to obtain a reasonable scheduling scheme for transferring the satellite. Hence, Δv and t_m can be considered as decision variables of the optimization problem, and they are constrained by

$$-\Delta v_{max} \leq \Delta vs. \leq \Delta v_{max}, \tag{25}$$

$$0 < t_m < t_t + T. \tag{26}$$

Since Δv is associated with the capacity of fuel, which is the key parameter of the satellite remained lifetime, constraint (25) ensures that the velocity increment is limited to a reasonable range. Here Δv_{max} represents the maximum allowed velocity increment and the negative value indicates the velocity in the reverse direction. Constraint (26) defines the range of a feasible maneuver moment (when the satellite starts its rocket engine). Furthermore, there are other constraints introduced in the following.

$$D_{imag}/10^6 \leq R, \tag{27}$$

$$250 \times 10^3 \leq H_{new} \leq 1300 \times 10^3, \tag{28}$$

$$t_s \leq t_t + T_r \leq t_e, \tag{29}$$

$$T_r \leq T. \tag{30}$$

To obtain sufficient information from a single observation result, constraint (27) guarantees that the ground resolution is smaller than the required resolution, in which the ground resolution is associated with the satellite altitude over the ground target divided by

the horizontal number of pixels. When the satellite is transferring its orbit, the change of orbit altitude should be limited to a reasonable range to ensure the stable operation of the satellite. As the satellite altitude is typically between 250 and 1300 km [43], constraint (28) is carried out to limit the satellite altitude after maneuvering. Constraint (29) is used to ensure that the observation moment lies in the sunshine time window. Furthermore, as timeliness is crucial for emergency observation tasks, we use constraint (30) to limit the maximum response time of the satellite.

In practical applications, users require different solutions depending upon the purpose. To satisfy diverse user requirements, we build three models considering response time, ground resolution, and fuel consumption as objectives, respectively, which are written as

$$\begin{aligned} f_3 &= min\ T_r, \\ s.t.\ &Constraints\ (25)–(29). \end{aligned} \tag{31}$$

$$\begin{aligned} f_2 &= min\left(D_{imag}/10^6\right), \\ s.t.\ &Constraints\ (25), (26), (28)–(30). \end{aligned} \tag{32}$$

$$\begin{aligned} f_1 &= min\ \Delta v, \\ s.t.\ &Constraints\ (26)–(30). \end{aligned} \tag{33}$$

The calculation processes of objectives are as follows. During the optimization process, appropriate decision variables (i.e., velocity vector increment Δv and maneuver moment t_m) will be searched by the algorithm under the constraints mentioned in Equations (31)–(33). Since the satellite conducts an impulsive maneuver whose direction and magnitude are determined by Δv at moment t_m to transfer its orbit, the new state velocity vector at moment t_m can be determined by decision variables. Then, the state velocity vector of the satellite can be transformed into the position vector represented by orbit elements by using Equations (19) and (20). As the satellite flies around the Earth, the position vector of the satellite changes with time. The changes in position vector can be tracked by Kepler's equation coupled with orbit equations mentioned in Equations (8)–(18). Meanwhile, according to the position vector, the subsatellite point at the same moment can be obtained by Equations (21)–(24). The FOV of the satellite is determined by the subsatellite point at the same moment according to Equations (1)–(7) introduced in the orbit coverage analysis. When a FOV covers the target point, it indicates that the satellite can observe this target point. Thus, the first objective f_1 is calculated by the difference between the maneuver time t_m and the time instance when the satellite can observe the target. In the second objective f_2, the satellite altitude is the distance between the satellite and the subsatellite point when the satellite can observe the target. As to the third objective f_3, it is determined by the decision variable Δv directly.

3. Adaptive Differential Evolution Algorithm Based on Graph Search

Differential evolution (DE) is an efficient population-based stochastic optimization approach for solving optimization problems over continuous space, and many variants of DE have been implemented in engineering fields [30,44,45]. In this study, we conduct problem-specific modifications on the framework of an adaptive DE, named ACODE, first proposed in [31] that concerned data clustering problems, to solve the orbit maneuver optimization problem. The ACODE can be treated as a hybridization of DE and ACO, which will be detailed in this section after a brief introduction to the classical DE.

3.1. Classical DE Algorithm

Typically, the DE includes four basic steps [46]: Initialization, mutation, crossover, and selection.

(i) Initialization. This step randomly creates an initial population consisting of N individuals. When the iteration number $G = 0$, the i-th individual is initialized in the

search space constrained by the minimum bound $X_{min} = \{x^1_{min}, x^2_{min}, \ldots, x^D_{min}\}$ and the maximum bound $X_{max} = \{x^1_{max}, x^2_{max}, \ldots, x^D_{max}\}$, according to the following method

$$\begin{aligned} &x^j_{i,0} = x^j_{min} + rand(0,1) \times (x^j_{max} - x^j_{min}), \\ &j \in \{1, 2, \ldots, D\}, \end{aligned} \tag{34}$$

where $rand(0,1)$ is a uniformly distributed number within $[0,1]$ and D is the number of dimensions.

(ii) Mutation. The mutation operation perturbs a target vector $X_{i,G}$ from the current generation to obtain a donor vector $V_{i,G}$, which can be written as

$$V_{i,G} = X_{r^i_1,G} + F \cdot (X_{r^i_2,G} - X_{r^i_3,G}), \tag{35}$$

where F is the scaling factor, which is a positive control parameter for scaling the difference vectors. The indices r^i_1, r^i_2, and r^i_3 are mutually exclusive integers randomly chosen from the range $[1, N]$ and they are different from the base vector index i.

(iii) Crossover. The crossover operation can improve the diversity of the population by exchanging the components of the donor vector $V_{i,G}$ with the target vector $X_{i,G}$ to form the trial vector $U_{i,G} = \left\{u^1_{i,G}, u^2_{i,G}, \ldots, u^D_{i,G}\right\}$. There are two kinds of commonly used crossover strategies, including exponential (i.e., two-point modulo) and binomial (i.e., uniform). The exponential crossover makes the trial vector contains a sequence of consecutive components taken from the parent vector. The structure of the trial vector can be expressed by

$$u^j_{i,G} = \begin{cases} v^j_{i,G}, & \text{if } j \in \{k, \langle k+1 \rangle_n, \ldots, \langle k+L-1 \rangle_n\}, \\ x^j_{i,G}, & \text{for all other } j \in [1, D]. \end{cases} \tag{36}$$

where $\langle j \rangle_n$ is a modulo function with modules D, k and L are two integers randomly chosen from $[1, D]$. On the other hand, the binomial strategy can be outlined as

$$u^j_{i,G} = \begin{cases} v^j_{i,G}, & \text{if } (rand_j[0,1] \leq CR \text{ or } j = j_{rand}), \\ x^j_{i,G}, & \text{otherwise.} \end{cases} \tag{37}$$

where CR is the crossover rate and j_{rand} is a randomly chosen index lying in the interval $[1, D]$.

(iv) Selection. The selection operation determines whether the target or the trial vector survives to the next generation according to the objective function, which is described as

$$X_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } f(U_{i,G}) \leq f(X_{i,G}), \\ X_{i,G}, & \text{otherwise.} \end{cases} \tag{38}$$

Once an initial population is created, the mutation, crossover, and selection strategies are repeated until a stopping criterion is satisfied to obtain promising solutions. It should be noted that different mutation strategies demarcate a DE scheme from other schemes. Except for the mutation strategy introduced above, there are some other well-known mutation strategies, such as "DE/best/1", "DE/best/2", "DE/rand/2", "DE/rand-to-best/1", "DE/current-to-pbest/1", "DE/current-to-rand/1", etc. [46].

3.2. ACODE Algorithm

The performance of DE highly depends on four key components, i.e., mutation strategy, crossover strategy, scaling factor F, and crossover rate CR [31]. We transform these components into a directed acyclic graph and implement an ant colony optimization-based adaptive DE algorithm to conduct the optimization process.

3.2.1. Directed Acyclic Graph Formed by Configurations

An example of the directed acyclic graph formed by configurations is shown in Figure 3. The graph includes five levels, of which a virtual point lies in the first level to gather all ants and the remaining four levels represent four key components (i.e., mutation strategy, crossover strategy, scaling factor F, and crossover rate CR), respectively. Every node in each level indicates a candidate configuration, and the nodes of two adjacent levels are fully connected. A path that starts from the first level and terminates at the fifth level can be treated as a combination of four candidate configurations, as the blue path in Figure 3 shows. Mathematically, the directed acrylic graph can be described by $\Phi = \{V, E\}$, where V is the set of nodes and $E \subseteq V \times V$ indicates directed arcs. The pheromone trail on the arcs connecting $v \in V$ to adjacent nodes in the next level is recorded by pheromone vector B_v. Hence, the length of B_v depends on the number of nodes in the next level. According to empirical considerations [31,33,46], the candidate configurations we used in this paper are summarized in Table 2.

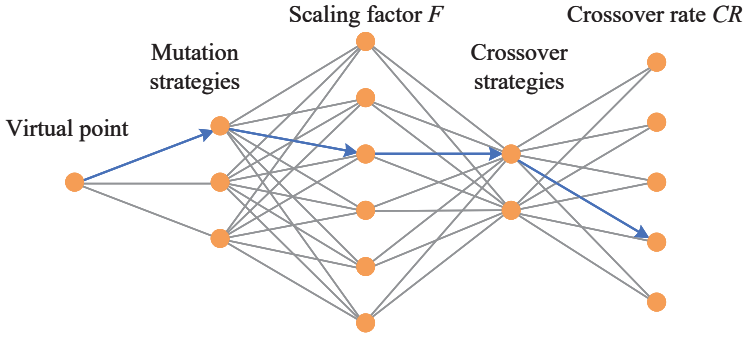


Figure 3. An example of the directed acyclic graph formed by key components of DE.

Table 2. Candidate configurations.

Components	Candidate Values or Strategies
Mutation strategy	"DE/rand/1", "DE/current-to-pbest/1", and "DE/current-to-rand/1"
Crossover strategy	binomial and exponential
Scaling factor F	0.4, 0.5, 0.6, 0.8, 0.9, and 1.1
Crossover rate CR	0.1, 0.4, 0.6, 0.9, and 0.99

3.2.2. Framework of ACODE

The framework of ACODE is displayed in Algorithm 1. The algorithm starts with the initialization of a population P with size N, configuration matrix M, iteration counter g, and pheromone matrix B^g (line 1). Particularly, the number of individuals in P equals the number of ants. Then, the algorithm runs until the stopping criterion is satisfied (lines 2–11). Every ant at each iteration is utilized to find a reasonable combination of configurations from the graph, and the combination is recorded in M (line 4). M_i indicates the configuration combination of the i-th individual and it is implemented to evolve individual $x_{i,g}$ (line 5). The offspring $u_{i,g}$ is compared with $x_{i,g}$ to determine which solution is preserved into the next generation (lines 6–9). Finally, the pheromone matrix B^{g+1} that would be used in the next generation is updated according to the fitness information in line 10.

3.2.3. Solution Representation and Initialization

As above-mentioned, we consider the velocity increment Δv and maneuver moment t_m as decision variables. During the optimization process, the decision variables are

used to calculate the positions of the satellite expressed by orbit elements, while the objectives are determined by position vectors, as introduced in Section 2. The velocity increment Δv is the magnitude of the change in the velocity vector, which can be represented by three velocities (i.e., Δv_x, Δv_y, and Δv_z) on three axes of the Cartesian coordinate system. Here the X axis is directed to the eccentricity vector, Z axis is in the direction of the satellite's angular momentum which lies perpendicular to the orbital plane, and the Y axis completes the right-hand set of co-ordinate axis. Therefore, a chromosome should be composed of velocity increments in three dimensions and the moment when the maneuver occurs. Figure 4 shows the representation of a chromosome, where x_i is the expression vector while t_m, Δv_x, Δv_y, and Δv_z are decision variables. Since the satellite conducts a coplanar maneuver, the velocity increment Δv_z always equals 0. Nevertheless, we still include Δv_z in the chromosome for computation convenience. In addition, we generate the initial population randomly and the boundaries of decision variables are determined by constraints (25) and (26). Since the search space of each variable is large and the performance of DE algorithm is seriously influenced by the diversity of the initial population, the initial population should be uniformly distributed in the search space. We use the Latin hypercube sampling (LHS) to generate the initial population. The LHS is a statistical method that can generate a quasi-random sampling distribution, which has been widely applied in other studies to obtain a high-quality initial population [47].

Algorithm 1: Framework of ACODE.

Input: Population size N, evaporation rate ρ, and directed acyclic graph Φ
Output: Final population P

Initialization: initial population $P \leftarrow \{x_1, x_2, \ldots, x_N\}$, configuration matrix $M \leftarrow \emptyset$, $g \leftarrow 0$, pheromone matrix $B^g \leftarrow \emptyset$,
while *stopping criterion is not satisfied* **do**
  for $i \in N$ **do**
    $M_i \leftarrow ParameterAdaption(\Phi, B^g)$
    $u_{i,g} \leftarrow GeneticOperation(M_i, x_{i,g})$
    if $f(u_{i,g}) < f(x_{i,g})$ **then**
      $x_{i,g+1} \leftarrow u_{i,g}$
    end
    else
      $x_{i,g+1} \leftarrow x_{i,g}$
    end
  end
  $B^{g+1} \leftarrow UpdatePheromone(B^g, \rho, P)$
  $g \leftarrow g + 1$
end

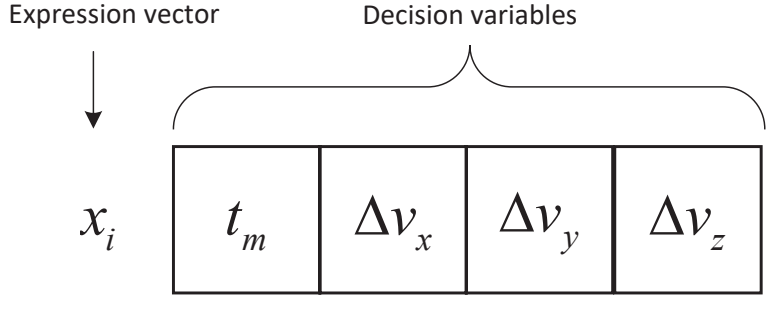


Figure 4. An illustration of the chromosome representation.

3.2.4. Parameters Adaption Based on ACO

Based on the directed acyclic graph, we conduct the parameter adaption by utilizing an ACO method. Specifically, each ant searches for a reasonable combination (i.e., path) for configuring an individual according to the pheromone trail on each arc in the graph. Given a node v from which an ant departs, there would be h candidate arcs that can be chosen. The probability p_j for picking arc j is written as

$$p_j = \frac{B_{v,j}^g}{\sum_{j=1}^{h} B_{v,j}^g}, \tag{39}$$

where $B_{v,j}^g$ is the pheromone trail on arc j with the starting node v at the g-th iteration. The process of parameter adaption is shown in Algorithm 2. In the algorithm, an ant departs from the virtual node v_1 and travels through four nodes in the remaining four levels. At the l-th level, all probabilities for choosing arcs connecting starting node v_l with all nodes in the next level are calculated based on Equation (39) and recorded in P_l (line 3). Then, roulette wheel selection (i.e., $RouletteWheel()$) is adopted to choose an arc j that determines the node v_{l+1} (i.e., end node of arc j) at the $(l+1)$-th level (lines 4–5). The roulette wheel selection is a well-known stochastic selection method, in which the probability for the selection of an arc is proportional to the pheromone trails on it. The above steps are repeated until a path consisting of four arcs is obtained. This algorithm is embedded into Algorithm 1 by executing once for each individual.

Algorithm 2: $ParameterAdoption()$.

Input: Directed acyclic graph Φ and pheromone matrix B^g
Output: configuration combination M_i
1 **Initialization:** $M_i \leftarrow \emptyset$, virtual node v_1
2 **for** $l \in [1, 4]$ **do**
3 $\quad P_l \leftarrow$Calculate probabilities of h arcs with Equation (39)
4 $\quad j \leftarrow RouletteWheel(P_l)$
5 $\quad$ Obtain node v_{l+1} according to arc j
6 $\quad M_i \leftarrow M_i \cup v_{l+1}$
7 **end**

3.2.5. Pheromone Update

In Algorithm 1, the pheromone trails of the whole graph at the generation g is recorded by a pheromone matrix B^g. The pheromone trail on each arc is updated at the end of each iteration by the following method

$$\Delta\tau_{v,j}^g = \frac{\sum_{x_t \in P_j^g} |f(x_{t,g+1}) - f(x_{t,g})|}{\sum_{i=1}^{N} |f(x_{t,g+1}) - f(x_{t,g})|}, \tag{40}$$

$$B_{v,j}^{g+1} = (1-\rho) \cdot B_{v,j}^g + \Delta\tau_{v,j}^g, \tag{41}$$

where $\Delta\tau_{v,j}^g$ is the pheromone increment on arc j with starting node v at the g-th iteration, P_j^g is the set of individuals who use the configurations corresponding to arc j at the g-the iteration, and ρ is the evaporation rate. Equation (40) indicates that the pheromone increment on an arc is determined by accumulated fitness improvements of individuals who passed this arc divided by that of all individuals. The pheromone trail $B_{v,j}^{g+1}$ on arc j with starting node v at the $(g+1)$-th iteration is updated by pheromone increment $\Delta\tau_{v,j}^g$ and pheromone trail $B_{v,j}^g$ at the g-the iteration, as well as evaporation rate ρ in Equation (41). Further, to avoid premature convergence, the Max–Min ant system [48] is implemented in this study to limit the pheromone level on each arc within a range $[0.1, 0.9]$.

4. Computational Experiments

To demonstrate the efficiency of ACODE on the proposed problem, simulation experiments are conducted in this section. All algorithms are coded in Matlab and run on a 64-bit Windows OS with Intel Core(TM) i5-8265U, 1.6 GHz, and 8 GB RAM.

4.1. Scenario Settings

We assume a set of scenarios in which three satellites are requested to observe four ground targets within 12 h (from 1 December 2020 14:00:00 to 2 December 2020 02:00:00, Beijing time). At the initial moment (i.e., 1 December 2020 14:00:00), each ground target is invisible to each satellite. Once an observation task is received, an appropriate satellite would be selected from these satellites to undertake orbital maneuvers to accomplish the task according to users' requirements. The initial orbital elements are displayed in Table 3, where the first column is the satellite ID and the other columns indicate semimajor axis a, inclination i, right ascension of the ascending node Ω, eccentricity e, argument of perigee ω, and mean anomaly M. The four ground targets are randomly located in low-latitude, mid-latitude, high-latitude, and higher-latitude areas, and their geographical information is summarized in Table 4. The maximum scanning angle of the satellite, maximum response time, minimum ground resolution, and maximum velocity increment predefined by users are set to 45°, 12 h, 2 m, and 300 m/s, respectively. Moreover, the maximum number of fitness evaluations (FEs) of the algorithm is set to 50,000 and the evaporation rate ρ is set to 0.8 according to pre-experiments.

Table 3. Initial orbital elements of satellites.

ID	a (m)	e	i (rad)	Ω (rad)	ω (rad)	M (rad)
1	6,878,140	3.59426×10^{-16}	97.0346	250.884	0	0
2	6,878,140	4.55556×10^{-18}	97.0346	10.8840	0	2.61014×10^{-16}
3	6,878,140	1.79873×10^{-16}	97.0346	130.884	0	5.08063×10^{-15}

Table 4. Geographical information of ground targets.

Target ID	Latitude	Longitude
1	0°	62°W
2	41°N	70°E
3	50°S	146°W
4	45°N	116°E

4.2. Simulation Results

The ACODE is implemented to solve the three optimization models with different optimization objectives based on the generated scenarios. The experiment results are summarized in Table 5, in which the columns indicate scenarios, satellites selected to accomplish observation tasks, maneuver moment t_m, velocity increments (Δv_x and Δv_y), and objective values (f_1, f_2, and f_3) of the three optimization models. Particularly, the scenario index is composed of the ground target ID and optimization objective. For instance, T1O1 means in this scenario the satellites are requested to observe ground target 1 and the optimization objective is f_1 involved by the first optimization model. Here f_1, f_2, and f_3 are response time, ground resolution, and fuel consumption, respectively. Note that the minimum fuel consumption is represented by minimum velocity increment, as discussed in Section 2. Although only one objective is considered in each scenario, we provide the values of the other two objectives corresponding to the optimal solution of the scenario. The value of the optimized objective considered in each scenario is in **boldface**.

The results in Table 5 indicate that all scenarios can be well-addressed by ACODE. Furthermore, it can be observed that huge differences in objective values can be obtained if we execute the same observation task based on different optimization models. For example,

T1O1, T1O2, and T1O3 are three scenarios in which the satellites are requested to observe ground target 1 with three different optimization objectives, respectively. The solution of T1O1 selects satellite 3 to execute the task and earns the minimum response time while yielding the poorest ground resolution compared with solutions of T1O2 and T1O3. More specifically, the solution of T1O1 decreases the response time by up to 84.44% and increases the ground resolution by up to 200.12% compared with the solutions of T1O2 and T1O3. Meanwhile, its fuel consumption is a little less than the solution of T1O2 that optimizes the ground resolution and much more than the solution of T1O3 that aims to find the minimum fuel consumption.

Table 5. Simulation results.

Scenario	Selected Satellite ID	t_m	Δv_x (m/s)	Δv_y (m/s)	f_1 (s)	f_2 (m)	f_3 (m/s)
T1O1	3	2020-12-1 14:49	−81.744079	288.2563	**5682** [1]	1.34	299.62
T1O2	1	2020-12-1 14:10	286.203453	89.93099	36507	**0.43**	300.00
T1O3	1	2020-12-1 14:45	−65.113285	−9.637086	36507	1.08	**65.82**
T2O1	1	2020-12-1 14:59	123.578258	272.2139	**6301**	1.35	298.95
T2O2	3	2020-12-1 14:46	−155.259419	−256.6993	20440	**0.44**	300.00
T2O3	3	2020-12-1 14:00	3.676842	0.506845	19243	0.74	**3.71**
T3O1	2	2020-12-1 14:44	−224.161983	198.8964	**4833**	1.41	299.68
T3O2	2	2020-12-1 15:10	−252.270809	162.3559	40115	**0.44**	300.00
T3O3	1	2020-12-1 14:11	15.541757	−15.38881	9401	0.85	**21.87**
T4O1	3	2020-12-1 15:14	293.379811	−56.7	**7705**	1.01	298.81
T4O2	1	2020-12-1 14:45	−137.707437	−266.5268	38606	**0.43**	300.00
T4O3	1	2020-12-1 15:20	24.506321	31.81572	7859	0.95	**40.16**

[1] The values in **boldface** are optimized objectives in each scenario.

4.3. Algorithm Comparisons

To further demonstrate the superiority of ACODE, we compare it with three well-known evolutionary algorithms in existing studies, i.e., EPSDE [33], CSO [34], and SLPSO [35]. Particularly, EPSDE is an ensemble-based DE algorithm, in which a pool of mutation strategies along with a pool of corresponding control parameters compete to produce offspring individuals. CSO is a competitive swarm optimizer inspired by particle swarm optimization. In CSO, a pairwise competition mechanism is used to update the position of the particle that loses the competition by learning from the winner. Similarly, SLPSO adopts social learning mechanisms for particle swarm optimization. Meanwhile, a dimension-dependent parameter control method is embedded into the SLPSO to ease the burden of parameter settings.

The comparison results are summarized in Table 6, in which the last four columns are best, worst, mean, and standard deviation values of three optimization objectives over 10 runs obtained by all algorithms. Note that the fuel consumption is represented by the value of velocity increment. For each scenario, the best results are in **boldface**. Wilcoxon rank-sum tests with a significance level of 0.05 are used for the significance tests. It can be found that ACODE significantly outperforms EPSDE, CSO, and SLPSO in almost all scenarios, in terms of response time, ground resolution, and fuel consumption. Especially, the superiority of ACODE is more significant when it optimizes orbital maneuvers for observing ground target 1 in terms of response time (scenario T1O1) and ground target 2 in terms of fuel consumption (scenario T2O3).

Table 6. Algorithm comparison results.

Scenario	Algorithm	Best	Worst	Mean	Std.
T1O1	ACODE	**5682** [1]	**5682**	**5682**	**0**
	EPSDE	5689	5721	5701.8	9.71
	CSO	5689	19,935	7124.8	4270.07
	SLPSO	5693	19,909	7194.9	4239.05
T1O2	ACODE	**0.43**	**0.43**	**0.43**	**0**
	EPSDE	**0.43**	0.47	0.45	0.01
	CSO	0.48	0.64	0.56	0.06
	SLPSO	0.49	0.63	0.55	0.04
T1O3	ACODE	**65.82**	**65.82**	**65.82**	**0**
	EPSDE	68.18	76.5	71.68	3.15
	CSO	67.34	80.74	74.85	3.6
	SLPSO	76.98	120.14	97.47	12.66
T2O1	ACODE	**6301**	**6301**	**6301**	**0**
	EPSDE	6308	6341	6324.5	10.76
	CSO	6311	6339	6322.5	9.11
	SLPSO	6355	19,146	10,222	5834.61
T2O2	ACODE	**0.44**	**0.44**	**0.44**	**0**
	EPSDE	0.45	0.47	0.46	0.01
	CSO	0.47	0.55	0.51	0.03
	SLPSO	0.44	0.58	0.5	0.04
T2O3	ACODE	**3.71**	**3.81**	**3.72**	**0.03**
	EPSDE	6.31	10.06	8.28	1.3
	CSO	4.78	11.89	6.95	2.24
	SLPSO	9.4	72.13	29.45	18.07
T3O1	ACODE	**4833**	**4833**	**4833**	**0**
	EPSDE	4848	4882	4858.4	9.77
	CSO	4888	5129	4944.3	66.58
	SLPSO	4843	9332	6236.6	2006.62
T3O2	ACODE	**0.44**	**0.44**	**0.44**	**0**
	EPSDE	0.45	0.51	0.47	0.02
	CSO	0.49	0.54	0.51	0.01
	SLPSO	0.46	0.58	0.5	0.04
T3O3	ACODE	**21.87**	**21.87**	**21.87**	**0**
	EPSDE	22.83	32.5	29.11	3.43
	CSO	23.93	29.52	27.11	1.68
	SLPSO	34.74	64.22	45.18	8.49
T4O1	ACODE	**7705**	**7705**	**7705**	**0**
	EPSDE	7723	7746	7732.7	7.4
	CSO	7726	7754	7738	8.06
	SLPSO	7769	7846	7804.1	24.92
T4O2	ACODE	**0.43**	**0.44**	**0.43**	**0**
	EPSDE	0.45	0.48	0.46	0.01
	CSO	0.47	0.6	0.51	0.04
	SLPSO	0.48	0.55	0.52	0.02
T4O3	ACODE	**40.16**	**40.16**	**40.16**	**0**
	EPSDE	42.91	46.58	45.07	1.16
	CSO	41.05	48.92	43.91	2.58
	SLPSO	45.07	74.64	60.88	9.85

[1] The values in **boldface** are the best results in each scenario.

It should be noted that EPSDE is similar to ACODE, as EPSDE ensembles a set of mutation strategies and corresponding control parameters in DE. Hence, EPSDE shows similar performance compared with ACODE for observing ground target 1 in terms of ground resolution (scenario T1O2). Nevertheless, ACODE is superior to EPSDE in other scenarios. The reasons can be twofold. First, EPSDE only ensembles mutation strategies and corresponding control parameters, while crossover strategies and corresponding control parameters that also can affect algorithm performance are not involved. On the contrary, ACODE considers both mutation strategies, crossover strategies, and their control parameters. Second, each component of EPSDE conducts the adaption independently while ACODE configures all components in a holistic manner, which is also the difference between ACODE and ensemble-based algorithms.

4.4. Experiments with Insufficient Satellite Resources

In the above sections, we calculate the solution of every satellite and select the most appropriate satellite out of three satellites to observe the ground target. The simulation results are obtained by the algorithm with sufficient satellite resources. However, since some satellites may be occupied by other tasks that cannot be interrupted when emergencies occur (i.e., some satellites may be infeasible for executing the observation task), it is interesting to investigate the impact of insufficient satellite resources on the orbital maneuver scheme and algorithm performance. Hence, this section analyzes the experiment results with different numbers of satellites based on the scenarios generated by removing satellites from the scenarios introduced in Section 4.1. Specifically, the one-satellite scenarios in this section preserve satellite 1, the two-satellite scenarios preserve satellite 1 and satellite 2, and the three-satellite scenarios are the same as before.

The simulation results are presented in Figure 5. Each figure indicates the simulation results for observing the same ground target, and the same color means the simulation results in the scenarios that consider the same optimization model. Moreover, to understand the trade-off among three objectives, we normalize all results into $[0, 1]$, and a smaller value indicates a better solution in a direction. Since we generate the scenarios by removing satellites from the scenarios that already have solutions in Section 4.2, some scenarios would have the same solution as before. For example, the solution schemes of three scenarios that observe target 1 while optimizing the ground resolution with different numbers of satellites select satellite 1 to execute the task. Hence, the results of scenarios T1O2-one-satellite, T1O2-two-satellite, and T1O2-three-satellite are the same, as Figure 5a shows. On the other hand, other solutions indicate that the number of satellites significantly affects the algorithm results. For example, scenarios T1O1-one-satellite, T1O1-two-satellite, and T1O1-three-satellite select three different satellites to execute the task, respectively. To observe ground target 1 with the aim of optimizing response time, satellite 2 is selected in the two-satellite scenario and the response time is increased by 249.93% compared with the solution of the three-satellite scenario, as Figure 5a shows.

Furthermore, it can be found that with the increase in the number of satellites, the value of the optimization objective that corresponds to each optimization model can be significantly improved. However, the trade-off results among the three objectives show that the improvement on one objective may not always promote the improvement of other objectives. For example, the ground resolution for observing ground target 2 is significantly improved as the number of satellites increases from 1 to 3, while the fuel consumption is still very high and the response time is even increased, as Figure 5b shows.

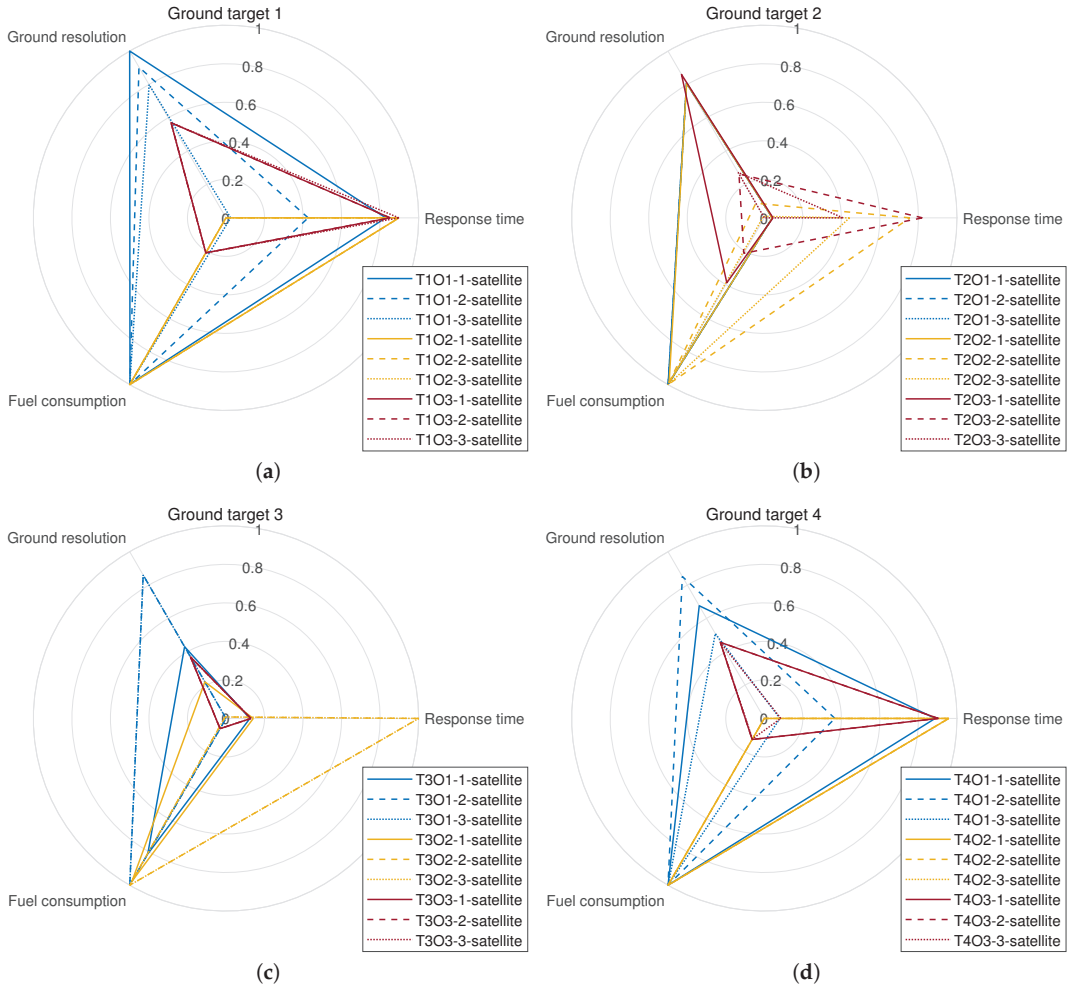


Figure 5. Simulation results by varying the number of satellite. (**a**) Ground target 1, (**b**) ground target 2, (**c**) ground target 3, (**d**) ground target 4.

5. Conclusions

In this paper, we investigate the orbital maneuver optimization problem of Earth observation satellites oriented to emergency tasks. Based on the analysis of orbit coverage and dynamics, we propose three kinds of optimization models that aim to, respectively, optimize response time, ground resolution, and fuel consumption, to satisfy diverse user requirements. Meanwhile, we implement an adaptive differential evolution algorithm based on graph search to solve the proposed optimization problems, which is named ACODE. The main feature of ACODE is to form the key components of DE into a directed acyclic graph and adopt an ACO method to search for combinations of these components from the graph, thereby adaptively configuring reasonable components for DE. The key components considered in this paper include mutation strategies, crossover strategies, as well as their corresponding control parameters, both of which can affect the performance of DE.

Finally, computational experiments are conducted to verify the proposed three optimization models and ACODE. The simulation results show that all simulation scenarios

that consider different optimization objectives can be well-addressed by ACODE. Comparison experiments are also carried out to demonstrate the superiority of ACODE on the proposed problem. The comparison results indicate that ACODE is superior to three well-known algorithms (i.e., EPSDE, CSO, and SLPSO). Further, we find that insufficient satellite resources would affect the efficiency of the orbital maneuver scheme and algorithm.

In future studies, we would like to investigate the multi-objective optimization algorithm that can optimize the three optimization objectives simultaneously for better decision making operations.

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

Funding: This research was jointly funded by the Natural Science Foundation of Hunan Province (No. 2021JJ30847) and China Scholarship Council (No. 202006370285).

Data Availability Statement: Not applicable.

Acknowledgments: We thank Anupam Trivedi for helping us improve the writing.

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

References

1. Wang, X.; Wu, G.; Xing, L.; Pedrycz, W. Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions. *IEEE Syst. J.* **2020**, *15*, 3881–3892. [CrossRef]
2. Verhegghen, A.; Kuzelova, K.; Syrris, V.; Eva, H.; Achard, F. Mapping Canopy Cover in African Dry Forests from the Combined Use of Sentinel-1 and Sentinel-2 Data: Application to Tanzania for the Year 2018. *Remote Sens.* **2022**, *14*, 1552. [CrossRef]
3. Chen, J.; Tang, H.; Ge, J.; Pan, Y. Rapid Assessment of Building Damage Using Multi-Source Data: A Case Study of April 2015 Nepal Earthquake. *Remote Sens.* **2022**, *14*, 1358. [CrossRef]
4. Chen, Y.; Mahalec, V.; Chen, Y.; He, R.; Liu, X. Optimal Satellite Orbit Design for Prioritized Multiple Targets with Threshold Observation Time Using Self-Adaptive Differential Evolution. *J. Aerosp. Eng.* **2015**, *28*, 04014066. [CrossRef]
5. Savitri, T.; Kim, Y.; Jo, S.; Bang, H. Satellite Constellation Orbit Design Optimization with Combined Genetic Algorithm and Semianalytical Approach. *Int. J. Aerosp. Eng.* **2017**, *2017*, 1235692. [CrossRef]
6. Sengupta, P.; Vadali, S.R.; Alfriend, K.T. Satellite Orbit Design and Maintenance for Terrestrial Coverage. *J. Spacecr. Rocket.* **2010**, *47*, 177–187. [CrossRef]
7. Graham, K.F.; Rao, A.V. Minimum-Time Trajectory Optimization of Low-Thrust Earth-Orbit Transfers with Eclipsing. *J. Spacecr. Rocket.* **2016**, *53*, 289–303. [CrossRef]
8. Wang, Z.; Grant, M.J. Optimization of Minimum-Time Low-Thrust Transfers Using Convex Programming. *J. Spacecr. Rocket.* **2017**, *55*, 586–598. [CrossRef]
9. Zhang, C.; Topputo, F.; Bernelli-Zazzera, F.; Zhao, Y.S. Low-Thrust Minimum-Fuel Optimization in the Circular Restricted Three-Body Problem. *J. Guid. Control Dyn.* **2015**, *38*, 1501–1510. [CrossRef]
10. Sadegh Mohammadi, M.; Naghash, A. Robust optimization of impulsive orbit transfers under actuation uncertainties. *Aerosp. Sci. Technol.* **2019**, *85*, 246–258. [CrossRef]
11. Cheng, L.; Wang, Z.; Jiang, F.; Zhou, C. Real-Time Optimal Control for Spacecraft Orbit Transfer via Multiscale Deep Neural Networks. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 2436–2450. [CrossRef]
12. Morante, D.; Sanjurjo-Rivo, M.; Soler, M.; Sánchez-Pérez, J.M. Hybrid multi-objective orbit-raising optimization with operational constraints. *Acta Astronaut.* **2020**, *175*, 447–461. [CrossRef]
13. Song, Z.; Chen, X.; Luo, X.; Wang, M.; Dai, G. Multi-objective optimization of agile satellite orbit design. *Adv. Space Res.* **2018**, *62*, 3053–3064. [CrossRef]
14. Appel, L.; Guelman, M.; Mishne, D. Optimization of satellite constellation reconfiguration maneuvers. *Acta Astronaut.* **2014**, *99*, 166–174. [CrossRef]
15. Paek, S.W.; Kim, S.; de Weck, O. Optimization of Reconfigurable Satellite Constellations Using Simulated Annealing and Genetic Algorithm. *Sensors* **2019**, *19*, 765. [CrossRef]
16. Sarno, S.; Guo, J.; D'Errico, M.; Gill, E. A guidance approach to satellite formation reconfiguration based on convex optimization and genetic algorithms. *Adv. Space Res.* **2020**, *65*, 2003–2017. [CrossRef]
17. McGrath, C.N.; Macdonald, M. General Perturbation Method for Satellite Constellation Reconfiguration Using Low-Thrust Maneuvers. *J. Guid. Control Dyn.* **2019**, *42*, 1676–1692. [CrossRef]

18. Soleymani, M.; Fakoor, M.; Bakhtiari, M. Optimal mission planning of the reconfiguration process of satellite constellations through orbital maneuvers: A novel technical framework. *Adv. Space Res.* **2019**, *63*, 3369–3384. [CrossRef]
19. He, X.; Li, H.; Yang, L.; Zhao, J. Reconfigurable Satellite Constellation Design for Disaster Monitoring Using Physical Programming. *Int. J. Aerosp. Eng.* **2020**, *2020*, 8813685. [CrossRef]
20. Wang, X.; Zhang, H.; Bai, S.; Yue, Y. Design of agile satellite constellation based on hybrid-resampling particle swarm optimization method. *Acta Astronaut.* **2021**, *178*, 595–605. [CrossRef]
21. Hu, J.; Huang, H.; Yang, L.; Zhu, Y. A multi-objective optimization framework of constellation design for emergency observation. *Adv. Space Res.* **2021**, *67*, 531–545. [CrossRef]
22. Pontani, M.; Conway, B.A. Particle swarm optimization applied to impulsive orbital transfers. *Acta Astronaut.* **2012**, *74*, 141–155. [CrossRef]
23. Zhang, S.; Duan, H. Gaussian pigeon-inspired optimization approach to orbital spacecraft formation reconfiguration. *Chin. J. Aeronaut.* **2015**, *28*, 200–205. [CrossRef]
24. dos Santos, D.P.S.; da Silva Formiga, J.K. Application of a genetic algorithm in orbital maneuvers. *Comput. Appl. Math.* **2015**, *34*, 437–450. [CrossRef]
25. Shirazi, A. Analysis of a hybrid genetic simulated annealing strategy applied in multi-objective optimization of orbital maneuvers. *IEEE Aerosp. Electron. Syst. Mag.* **2017**, *32*, 6–22. [CrossRef]
26. Yao, W.; Luo, J.; Macdonald, M.; Wang, M.; Ma, W. Improved Differential Evolution Algorithm and Its Applications to Orbit Design. *J. Guid. Control Dyn.* **2018**, *41*, 936–943. [CrossRef]
27. Hitomi, N.; Selva, D. Constellation optimization using an evolutionary algorithm with a variable-length chromosome. In Proceedings of the 2018 IEEE Aerospace Conference, Big Sky, MT, USA, 3–10 March 2018; pp. 1–12. [CrossRef]
28. Wu, G.; Mallipeddi, R.; Suganthan, P.N. Ensemble strategies for population-based optimization algorithms—A survey. *Swarm Evol. Comput.* **2019**, *44*, 695–711. [CrossRef]
29. Mallipeddi, R.; Wu, G.; Lee, M.; Suganthan, P.N. Gaussian adaptation based parameter adaptation for differential evolution. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 1760–1767. [CrossRef]
30. Wu, G.; Mallipeddi, R.; Suganthan, P.N.; Wang, R.; Chen, H. Differential evolution with multi-population based ensemble of mutation strategies. *Inf. Sci.* **2016**, *329*, 329–345. [CrossRef]
31. Wu, G.; Peng, W.; Hu, X.; Wang, R.; Chen, H. Configuring differential evolution adaptively via path search in a directed acyclic graph for data clustering. *Swarm Evol. Comput.* **2020**, *55*, 100690. [CrossRef]
32. Chen, Y.; Mahalec, V.; Chen, Y.; Liu, X.; He, R.; Sun, K. Reconfiguration of satellite orbit for cooperative observation using variable-size multi-objective differential evolution. *Eur. J. Oper. Res.* **2015**, *242*, 10–20. [CrossRef]
33. Mallipeddi, R.; Suganthan, P.; Pan, Q.; Tasgetiren, M. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl. Soft Comput.* **2011**, *11*, 1679–1696. [CrossRef]
34. Cheng, R.; Jin, Y. A Competitive Swarm Optimizer for Large Scale Optimization. *IEEE Trans. Cybern.* **2015**, *45*, 191–204. [CrossRef] [PubMed]
35. Cheng, R.; Jin, Y. A social learning particle swarm optimization algorithm for scalable optimization. *Inf. Sci.* **2015**, *291*, 43–60. [CrossRef]
36. Vandenrijt, J.F. Simulation and graphical representation of the orbit and the imaging parameter of Earth observation satellites. *Acta Astronaut.* **2005**, *57*, 186–196. [CrossRef]
37. Zhu, K.J.; Li, J.F.; Baoyin, H.X. Satellite scheduling considering maximum observation coverage time and minimum orbital transfer fuel cost. *Acta Astronaut.* **2010**, *66*, 220–229. [CrossRef]
38. Vallado, D.A. *Fundamentals of Astrodynamics and Applications*; Springer: Berlin/Heidelberg, Germany, 2001; Volume 12.
39. Dong, Y.; Wei, X.; Tian, L.; Liu, F.; Xu, G. A Novel Double Cluster and Principal Component Analysis-Based Optimization Method for the Orbit Design of Earth Observation Satellites. *Int. J. Aerosp. Eng.* **2017**, *2017*, 6396032. [CrossRef]
40. Buzzi, P.G.; Selva, D.; Hitomi, N.; Blackwell, W.J. Assessment of constellation designs for earth observation: Application to the TROPICS mission. *Acta Astronaut.* **2019**, *161*, 166–182. [CrossRef]
41. Curtis, H. *Orbital Mechanics for Engineering Students*; Butterworth-Heinemann: Oxford, UK, 2013.
42. Edlund, E.M. Interception and rendezvous: An intuition-building approach to orbital dynamics. *Am. J. Phys.* **2021**, *89*, 559–566. [CrossRef]
43. Somma, G.L.; Lewis, H.G.; Colombo, C. Sensitivity analysis of launch activities in Low Earth Orbit. *Acta Astronaut.* **2019**, *158*, 129–139. [CrossRef]
44. Biswas, P.P.; Suganthan, P.N.; Wu, G.; Amaratunga, G.A.J. Parameter estimation of solar cells using datasheet information with the application of an adaptive differential evolution algorithm. *Renew. Energy* **2019**, *132*, 425–438. [CrossRef]
45. Wu, G.; Shen, X.; Li, H.; Chen, H.; Lin, A.; Suganthan, P.N. Ensemble of differential evolution variants. *Inf. Sci.* **2018**, *423*, 172–186. [CrossRef]
46. Das, S.; Suganthan, P.N. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Trans. Evol. Comput.* **2011**, *15*, 4–31. [CrossRef]

47. Zhao, Z.; Yang, J.; Hu, Z.; Che, H. A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric Latin hypercube design for unconstrained optimization problems. *Eur. J. Oper. Res.* **2016**, *250*, 30–45. [CrossRef]
48. Stützle, T.; Hoos, H.H. MAX–MIN Ant System. *Future Gener. Comput. Syst.* **2000**, *16*, 889–914. [CrossRef]

remote sensing

MDPI

Article

A Method for Digital Terrain Reconstruction Using Longitudinal Control Lines and Sparse Measured Cross Sections

Yunwen Pan [1,2], Junqiang Xia [1] and Kejun Yang [2,*]

[1] State Key Laboratory of Water Resources and Hydropower Engineering Science, Wuhan University, Wuhan 430072, China; panyunwen@whu.edu.cn (Y.P.); xiajq@whu.edu.cn (J.X.)
[2] State Key Laboratory of Hydraulics and Mountain River Engineering, Sichuan University, Chengdu 610065, China
* Correspondence: yangkejun@scu.edu.cn

Abstract: Using longitudinal control lines and sparse measured cross sections with large spaces, a new method for quickly reconstructing digital terrains in natural riverways is presented. The longitudinal control lines in a natural riverway, mainly including the river boundaries, the thalweg, the dividing lines of floodplains and main channel, and the water edges, can be obtained by interpreting satellite images, remote sensing images or site surveys. Then, the longitudinal control lines are introduced into quadrilateral grid generation as auxiliary lines that can control longitudinal riverway trends and reflect transverse terrain changes. Then, by the equal cross-sectional area principle at the same water level, all measured cross sections are reasonably fitted. On the above basis, by virtue of the fitted cross-sectional data and the weighted distance method, the terrain interpolations along the longitudinal grid lines are conducted to obtain the elevation data of all grid nodes. Finally, according to the readable text formats of MIKE21 and SMS, the gridded digital terrain and connection information are output by computer programming to achieve good construction of the data exchange channels and fully exploit the special advantages of various software programs for digital terrain visualization and further utilization.

Keywords: natural riverways; digital terrain reconstruction; longitudinal control lines; sparse measured cross sections

Citation: Pan, Y.; Xia, J.; Yang, K. A Method for Digital Terrain Reconstruction Using Longitudinal Control Lines and Sparse Measured Cross Sections. *Remote Sens.* **2022**, *14*, 1841. https://doi.org/10.3390/rs14081841

Academic Editors: Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Received: 21 February 2022
Accepted: 5 April 2022
Published: 11 April 2022

1. Introduction

In river numerical simulations, whether the elevation values at grid nodes can truthfully reflect the riverway terrain is very important, as it directly determines the reliability of numerical simulation results [1–3]. In the process of riverway terrain reconstruction, to guarantee the truth and reasonability of the reconstruction results, many types of river numerical software are required to provide a large amount of terrain data covering the simulation areas and uniformly distributed. Due to the long terrain measurement period and large investment, under most situations, the method of laying uniform measured points for the whole riverway is not used in practical surveys. Generally, only the cross-sectional terrain data with large spaces are measured [4]. If entirely relying on the interpolation functions of conventional software to construct a riverway mode, the terrain accuracy will be low, and the accuracy and credibility of the numerical simulation results will be difficult to ensure. Therefore, it is necessary to explore a terrain reconstruction method for the sparsely measured cross-sectional data.

In the early studies, many scholars were keen to develop new methods. For example, Lin et al. [5] proposed a method of automatically reconstructing three-dimensional objects by a series of cross sections. Hardy [6] proposed the radial basis function interpolation algorithm, which has been widely used in many fields such as terrain modeling and digital

approximation, but as the number of sampled points increases, the solution speed of the radial basis function interpolation model greatly decreases. Targeting the deficiency of the radial basis function interpolation algorithm, some scholars presented the residual iteration method [7], quick multipolarization method [8] and partition of unity method [9], providing quick and accurate solutions for the radial basis function interpolation model from different aspects. Yokota et al. [10] conducted parallel processing of the radial basis function interpolation algorithm to improve the terrain reconstruction ability. Yang et al. [11] proposed a new method to achieve cross-sectional interpolation and evaluation deduction by curved surface interpolation technology. A weighted interpolation algorithm was presented by Wagner et al. [12] for reconstructing the cross-sectional profiles. Caviedes-Voullieme et al. [13] presented a new algorithm for generating missing information between the cross sections and the riverbed. Lebrenz and Bardossy [14] proposed a quantile kriging interpolation method, which needs to estimate variable distribution with time at the observed sites, as well as the marginal distribution in each predetermined time step. Due to the unremitting efforts of the above scholars, there are many terrain reconstruction methods. The proposal of a new method is obviously a breakthrough, but its calculational accuracy is also worthy of paying enough attention. Therefore, some scholars strictly compare the calculational accuracies of the existing terrain reconstruction methods. For instance, Weber [15] believed that the interpolation effects of the radial basis function algorithm and the inverse distance weighted method are basically the same. Kraus [16] thought that the interpolation effect of multilayer curved surface superposition is better than that of the binary higher-degree polynomial and spline functions. Zimmerman et al. [17] pointed out that, without consideration of the terrain type and sampling method, the interpolation effect of the kriging interpolation algorithm is better than that of the inverse distance weighted method. Gichamo et al. [18] found correcting the vertical bias of elevation points by a high-accuracy terrain model can considerably improve the cross-sectional obtainment. Andes et al. [19] thought that the rectilinear inverse distance weighting methodology is fairly feasible for cross-sectional interpolation. Determining the optimal algorithm is undoubtedly an effective means to improve the calculational accuracy, but a single method cannot meet all the needs of terrain reconstruction, because the calculational accuracy depends not only on the algorithm itself, but also on the geomorphic type and sampling density. Therefore, some scholars started to deeply discuss the effects of geomorphic type and sampling density on calculational accuracy. For example, based on regular discrete point data, by analyzing the influence of the geomorphic type, sampling density and interpolation algorithm on the regular grid digital terrain interpolation, Aguilar et al. [20] concluded that the effect of the geomorphic type on the digital terrain interpolation is largest, that of the sampling density is the second, and that of the interpolation algorithm is smallest. In addition, a few scholars focused on the application of terrain reconstruction result. For instance, Chen et al. [21] pointed out that the appropriate use of the interpolated cross sections can increase the precision of hydraulic river models. Florinsky et al. [22] analyzed the spatial distribution of soil properties by the regression analysis of topographic data. Applying elevation data and satellite remote sensing data, Sun et al. [23] established a riverway digital elevation model using a curved orthogonal grid and calculated the silt dash quantity of the riverway.

When the terrain data are relatively conventional, the measured point distribution is uniform and the sampling density is large, the calculational accuracies of many ready-made terrain reconstruction methods can meet the actual production requirements. However, for the terrain data in natural rivers, the situation is fairly different. The terrain changes of a natural riverway are neither isotropic nor completely anisotropic, but have very distinct transverse and longitudinal spatial tropisms. The spatial tropisms can be indirectly reflected by the longitudinal control lines such as the river boundaries, the thalweg, the dividing lines of floodplains and main channel, the water edges, etc. In other words, the longitudinal control lines have a function of controlling the longitudinal riverway trends and reflecting the transverse terrain changes. Nevertheless, this special function of the

longitudinal control lines has not been paid enough attention by the engineering surveyors for a long time. In all previous studies, the spatial tropisms between an interpolated point and the adjacent elevation points have not been fully considered when constructing the interpolation weights. Especially in a natural riverway where the transverse and longitudinal terrain changes present obviously different spatial tropisms, if the spatial tropisms between an interpolated point and the adjacent elevation points are not considered, the terrain interpolation accuracy will inevitably be affected when constructing the river digital model. Furthermore, for a natural riverway, there are only sparse measured cross-sectional data in most situations, which often does not meet the calculational requirements of many ready-made terrain reconstruction methods, let alone guarantee the reliabilities of the calculated results. For the above stated reasons, a new method for quickly reconstructing riverway digital terrains using longitudinal control lines and sparse measured cross sections with large spaces is presented.

2. Digital Terrain Reconstruction Method

Digital terrain reconstruction for a natural riverway refers to utilizing a small number of sparse measured cross-sectional data to generate the dense scatter terrain data of the target riverway by an interpolation method. The method presented in this paper also belongs to this case, and its specific process can be roughly divided into four stages (in Figure 1).

Stage 1—data preparation: the plane coordinate data of the longitudinal control lines and the measured cross-sectional terrain data must be prepared first.

Stage 2—riverway grid generation: the longitudinal control lines are introduced into the process of quadrilateral grid generation so that the generated grid can be well adapted to the riverway boundary changes, controlling the longitudinal riverway trends and reflecting the transverse terrain changes.

Stage 3—measured cross-sectional fitting: the measured cross-sectional terrain data are used to interpolate the intersections of the measured cross sections and the longitudinal grid lines by the equal cross-sectional area principle at a same water level and weighted distance method to reasonably fit the measured cross sections.

Stage 4—scatter digital terrain generation: using the generalized cross-sectional data and the weighted distance method, the elevation interpolations of all grid nodes are conducted along longitudinal grid lines to generate the whole riverway digital terrain. The detailed digital terrain reconstruction process can be seen in following contents.

2.1. Data Preparation

Before the digital terrain reconstruction for a natural riverway, the plane coordinate data of the longitudinal control lines and the measured cross-sectional terrain data must be prepared first.

The longitudinal control line data are actually a series of sequential plane control points, which can determine the plane shapes of the longitudinal control lines and can be obtained by interpreting satellite images, remote sensing images or site surveys [24–29]. As shown in Figure 2, for any a longitudinal control line, the first control point and the final control point are the intersections of the inlet measured cross section, the outlet measured cross section and the longitudinal control line. If the effects of river control works, production dikes and other engineering boundaries need to be considered, the positional coordinates of the engineering boundaries should be provided. In addition, It should be pointed out that this method does not require the acquisition of too many longitudinal control lines, but generally speaking, the more the number of the used longitudinal control lines is, the more accurate the calculated result is.

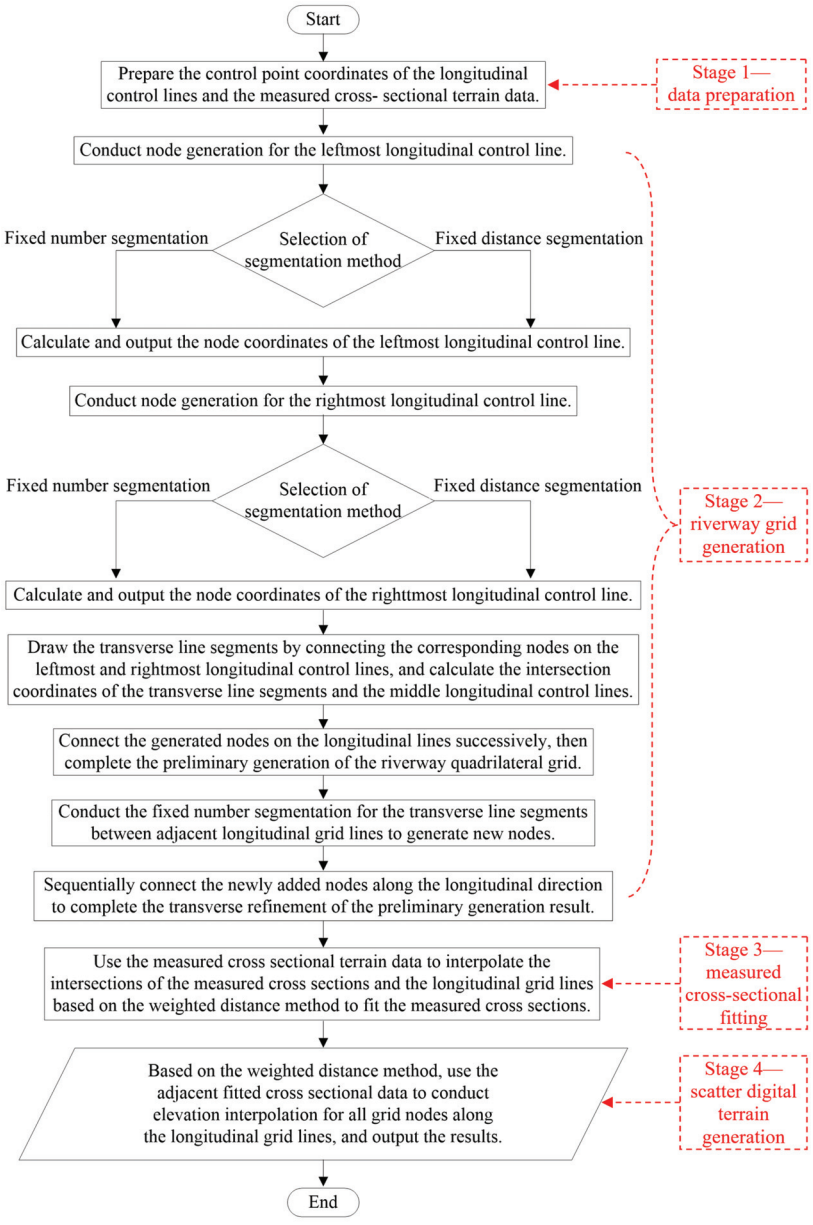


Figure 1. Digital terrain reconstruction process.

The measured cross-sectional terrain data mainly include the cross-sectional left and right endpoint coordinates (the left and right sides are defined according to the flow direction), the measured point elevations and the horizontal distances between the measured points and the corresponding left endpoint (in Figure 3).

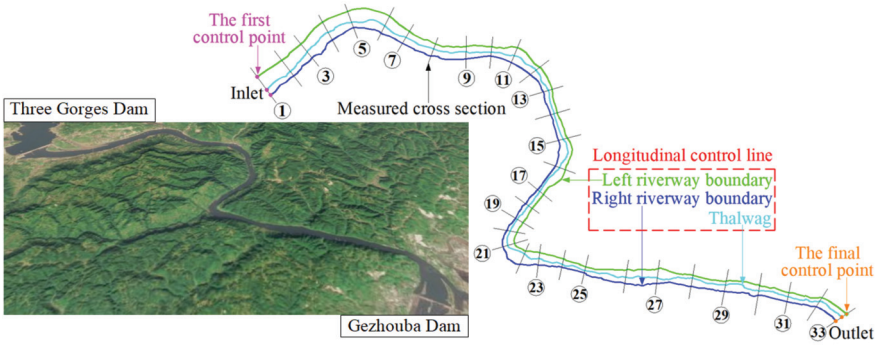


Figure 2. Distributions of longitudinal control lines and measured cross sections (The exemplified riverway is located between the Three Gorges Dam and Gezhouba Dam in China and approximately 30 km in length. The distances between the adjacent sections are approximately 600~1100 m).

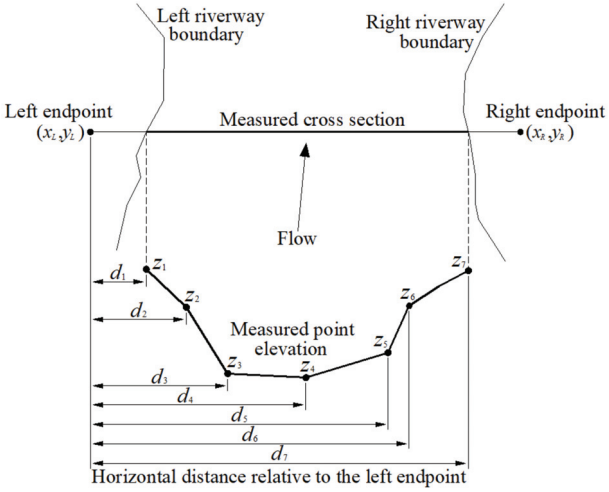


Figure 3. Schematic sketch of the measured cross-sectional data.

In this paper, a natural riverway, which is located between the Three Gorges Dam and Gezhouba Dam in China and approximately 30 km in length (in Figure 2), is used to briefly describe the digital terrain reconstruction process. In the early stage, 33 sparse measured cross sections and 3 longitudinal control lines (the left riverway boundary, the right riverway boundary and the thalweg) of the exemplified riverway were obtained through peer collection, visual judgment of satellite images and field survey, therein the distances between the adjacent measured cross sections are approximately 600~1100 m.

2.2. Riverway Grid Generation

In this paper, quadrilateral cells are used to conduct the grid generation for the target river reach. Riverway grid generation is an important process in the digital terrain reconstruction, and its specific process can be roughly divided into five steps.

Step 1—cumulative distance calculation for the riverway boundary control points.
Step 2—node generation for the left riverway boundary.
Step 3—node generation for the right riverway boundary.
Step 4—preliminary grid generation.
Step 5—transverse refinement for the preliminary grids.

2.2.1. Cumulative Distance Calculation for the Riverway Boundary Control Points

For generating riverway grids, the cumulative distances of the riverway boundary control points relative to the corresponding first control points should be firstly calculated. Assuming that the control point number of the left riverway boundary is N_L, if the cumulative distance of the first control point relative to itself is recorded as 0 and the serial number is 1, then the plane coordinates and cumulative distances of the control points on the left riverway boundary can be combined into $(x_L(i), y_L(i), L_L(i))$ where $i = 1, 2, 3, \ldots, N_L$, therein $L_L(1) = 0$. Similarly, assuming that the control point number of the right riverway boundary is N_R, then the plane coordinates and cumulative distances of the control points on the right riverway boundary can be combined into $(x_R(i), y_R(i), L_R(i))$ where $i = 1, 2, 3, \ldots, N_R$, therein $L_R(1) = 0$. Actually, for any a longitudinal control line, the cumulative distance of the final control point relative to the first control point is equal to the total length of the longitudinal control line, and the difference between two adjacent cumulative distances is the distance between the two corresponding control points.

2.2.2. Node Generation for the Left Riverway Boundary

After the cumulative distances of the boundary control points relative to the corresponding first control points are calculated and stored in the sequential data format, node generation could be conducted for the left riverway boundary. Here, the fixed number segmentation method or the fixed distance segmentation method may be selected.

Assuming the fixed number segmentation method is selected and n_L is taken as the segmentation number, then the generation step size s_L along the left river boundary is equal to $L_L(N_L)/n_L$, and finally $n_L + 1$ nodes could be generated. According to the cumulative distances, the left riverway boundary is divided into $N_L - 1$ cumulative distance intervals, namely $[L_L(1), L_L(2)]$, $[L_L(2), L_L(3)]$, …, $[L_L(N_L - 1), L_L(N_L)]$. Then, taking the first control point as the starting point, the stepping distances $j{\cdot}s_L$ $(j = 0, 1, 2, \ldots, n_L)$ along the left riverway boundary could be successively calculated. If a stepping distance $j{\cdot}s_L$ falls in the cumulative distance interval $[L_L(k), L_L(k+1)]$, namely $L_L(k) \le j{\cdot}s_L \le L_L(k+1)$, then the node coordinate $(x_{L,\mathrm{node}}(j+1), y_{L,\mathrm{node}}(j+1))$ can be calculated according to Formulas (1) and (2).

$$x_{\xi,\mathrm{node}}(j+1) = \frac{(j{\cdot}s_\xi - L_\xi(k)){\cdot}(x_\xi(k+1) - x_\xi(k))}{\sqrt{(x_\xi(k+1) - x_\xi(k))^2 + (y_\xi(k+1) - y_\xi(k))^2}} + x_\xi(k) \tag{1}$$

$$y_{\xi,\mathrm{node}}(j+1) = \frac{(j{\cdot}s_\xi - L_\xi(k)){\cdot}(y_\xi(k+1) - y_\xi(k))}{\sqrt{(x_\xi(k+1) - x_\xi(k))^2 + (y_\xi(k+1) - y_\xi(k))^2}} + y_\xi(k) \tag{2}$$

where the subscript ξ is equal to L or R; when $\xi = L$, the above formulas are used to calculate the node coordinates of the left river boundary; when $\xi = R$, the above formulas are used to calculate the node coordinates of the right river boundary.

Assuming the fixed distance segmentation method is selected and $L_{0,L}$ is taken as the segmentation distance, then the generation step size s_L along the left river boundary is equal to $L_{0,L}$, but the generated node amount is relevant to the segmentation distance $L_{0,L}$. When the remainder of $L_L(N_L)/L_{0,L}$ is equal to 0, the fixed distance segmentation method is equivalent to the fixed number segmentation method whose segmentation number n_L equals $L_L(N_L)/L_{0,L}$, and finally $L_L(N_L)/L_{0,L} + 1$ nodes could be generated, and the node coordinates $(x_{L,\mathrm{node}}(j+1), y_{L,\mathrm{node}}(j+1))$ $(j = 0, 1, 2, \ldots, L_L(N_L)/L_{0,L})$ can be successively calculated according to Formulas (1) and (2). When the remainder of $L_L(N_L)/L_{0,L}$ is not equal to 0, the segmentation number n_L equals $[L_L(N_L)/L_{0,L}] + 1$ where $[L_L(N_L)/L_{0,L}]$ represents that only the integral part of $L_L(N_L + 1)/L_{0,L}$ is used, and finally $[L_L(N_L)/L_{0,L}] + 2$ nodes could be generated. Therein, the first $[L_L(N_L)/L_{0,L}] + 1$ node coordinates $(x_{L,\mathrm{node}}(j+1), y_{L,\mathrm{node}}(j+1))$ $(j = 0, 1, 2, \ldots, [L_L(N_L)/L_{0,L}])$ can be successively calculated according to Formulas (1) and (2), and the final control point of the left riverway boundary is actually the final node.

2.2.3. Node Generation for the Right Riverway Boundary

The next step, node generation is conducted for the right riverway boundary. Here, the fixed number segmentation method or the fixed distance segmentation method may be selected. As the generated grid cells in the target river reach are quadrilateral, the segmentation number of the right riverway boundary must be equal to that of the left riverway boundary, namely $n_R = n_L$.

Assuming the fixed number segmentation method is selected and the segmentation number is n_R, then the generation step size s_R along the right river boundary is equal to $L_R(N_R)/n_R$, and finally $n_R + 1$ nodes could be generated. Here, the calculational method of the node coordinates likes the left river boundary when the fixed number segmentation method is selected. Only at this time, the subscript ξ in Formulas (1) and (2) is equal to R.

Assuming the fixed distance segmentation method is selected, then the segmentation distance $L_{0,R}$ must be within $[L_R(N_R)/n_R,\ L_R(N_R)/(n_R - 1)]$ to guarantee $n_R = n_L$, and the generation step size s_R is equal to $L_{0,R}$, and finally $n_R + 1$ nodes are generated. If $L_{0,R} = L_R(N_R)/n_R$, the calculational method of the node coordinates likes the left river boundary when the fixed distance segmentation method is selected and the remainder of $L_L(N_L)/L_{0,L}$ is equal to 0. If $L_R(N_R)/n_R < L_{0,R} < L_R(N_R)/(n_R - 1)$, the calculational method of the node coordinates likes the left river boundary when the fixed distance segmentation method is selected and the remainder of $L_L(N_L)/L_{0,L}$ is not equal to 0.

2.2.4. Preliminary Grid Generation

After the node generation of the riverway boundaries, the corresponding nodes on the left and right riverway boundaries are first connected along the transverse directions. Then, the intersection coordinates of the connected line segments and the thalweg are calculated to complete the node generation of the thalweg. Finally, the generated nodes are connected successively along the longitudinal direction. At this point, the preliminary quadrilateral grid generation in the target reach is completed. However, due to the small number of longitudinal control lines, the transverse spaces of the longitudinal grid lines are large (in Figure 4).

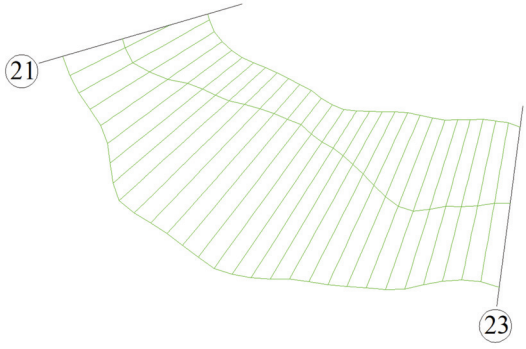


Figure 4. Preliminary generation result of quadrilateral grids (In the exemplified preliminary grid generation, the fixed number segmentation method is applied to the riverway boundaries).

2.2.5. Transverse Refinement for the Preliminary Grids

To decrease the transverse spaces between adjacent longitudinal grid lines, the transverse line segments between adjacent longitudinal grid lines must be subdivided. To ensure a uniform transverse density of the generated grids, the transverse line segments between any two adjacent longitudinal grid lines must be divided by the fixed number segmentation method. The segmentation numbers between different adjacent longitudinal grid lines can be different, but those between the same adjacent longitudinal grid lines must be the same, and the specific values should be determined according to the transverse spaces between the adjacent longitudinal grid lines. The larger the transverse spaces are, the larger the

segmentation numbers should be. Then, the newly added nodes are successively connected along the longitudinal direction. After the adding line operations are conducted in the regions between any two adjacent longitudinal grid lines, the transverse refinement of the preliminary generation result is completed (in Figure 5).

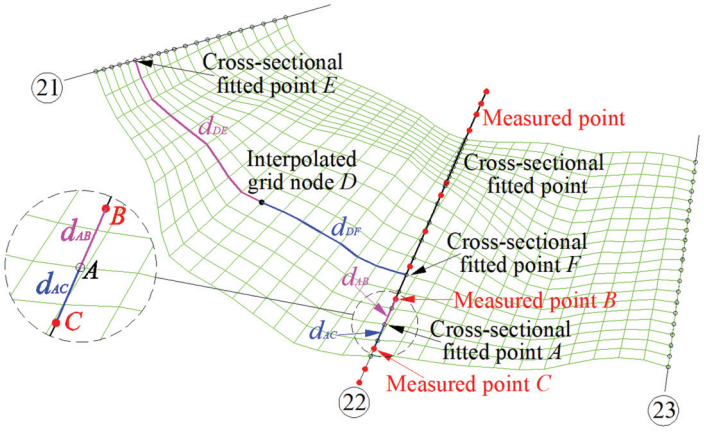


Figure 5. Transverse refinement effect of the preliminary grids and elevation interpolation schematic by the weighted distance method (In the process of the illustrative transverse refinement, the transverse segmentation numbers between the adjacent longitudinal control lines from the left side to right side are all 10).

2.3. Measured Cross-Sectional Fitting

After the grid generation in the target river reach is completed, the intersection coordinates of each measured cross section and the longitudinal grid lines could be calculated and translated into the horizontal distances relative to the measured cross-sectional left endpoint. Then, the measured cross-sectional data are used to interpolate the intersections of the measured cross sections and the longitudinal grid lines by the weighted distance method to reasonably fit the measured cross sections (Figure 6 only shows the 22nd cross section). The weighted distance method is an interpolation method based on the similarity principle, and it constructs the interpolation weights by the distances between the interpolated point and the sampled points. The shorter the distance between the interpolated point and a sampled point is, the greater the weight granted by the sampled point. In this step, the specific interpolation process is as follows: As shown in Figure 5, for a cross-sectional fitted point A and two measured points B and C, which are located in the same cross section as that of point A and are closest to point A, if the elevations of points B and C are z_B and z_C, and the distances between A and points B and C are, respectively, d_{AB} and d_{AC}, then the elevation z_A of point A can be calculated according to Formula (3).

$$z_A = d_{AC} \cdot z_B / (d_{AB} + d_{AC}) + d_{AB} \cdot z_C / (d_{AB} + d_{AC}) \quad (3)$$

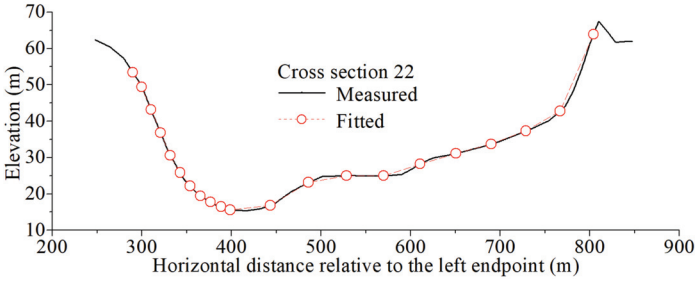


Figure 6. Measured cross-sectional fitting.

The rationality of measured cross-sectional fittings directly determines the accuracy of the reconstructed riverway digital terrain. The most ideal situation is that the fitted shapes of the measured cross sections are entirely consistent with the actual cross-sectional terrains. In other words, at th same water level, a fitted cross section and the corresponding measured cross section have same cross-sectional parameters (mainly including cross-sectional area, wetted perimeter, hydraulic radius, water surface width and average water depth), which is referred to as the equal cross-sectional area principle. The good or bad actual fitting effects are directly relevant to the selective rationality of the longitudinal control lines and the transverse spaces of the generated grids. When the selection of the longitudinal control lines in a target riverway is reasonable and the transverse spaces of the generated grids are small, the fitted cross sections are usually close to the actual cross-sectional terrain. If judging the rationality of the fitted cross sections is necessary, the cross-sectional parameters of the measured cross sections and the fitted cross sections under a series of water level conditions can be calculated for comparison.

2.4. Scatter Digital Terrain Generation

After the measured cross sections are reasonably fitted, the elevation interpolations of the grid nodes can be conducted along the longitudinal grid lines by the adjacent fitted cross sections and the weighted distance method, and the whole riverway digital terrain can be generated (in Figure 7). In this step, the specific interpolation process is as follows: As shown in Figure 5, assuming that the interpolated grid node is D, the two cross-sectional fitted points with known elevations and the closest longitudinal distances relative to point D are E and F, and point D is located in the middle of points E and F, and the elevations of points E and F are, respectively, z_E and z_F, and the longitudinal distances between D and points E and F are, respectively, d_{DE} and d_{DF}, then the elevation z_D of point D can be calculated according to Formula (4).

$$z_D = d_{DF} \cdot z_E / (d_{DE} + d_{DF}) + d_{DE} \cdot z_F / (d_{DE} + d_{DF}) \quad (4)$$

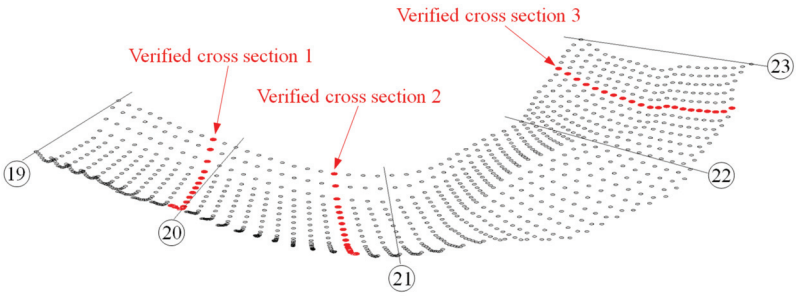


Figure 7. Riverway scattered digital terrain.

3. Comparisons between the Measured and Calculated Results

To verify the reasonableness of the calculated results, authors conducted a supplementary measurement for the actual terrains of the three verified cross sections (in Figure 7). Through a comparison of the calculated results with the measured results of the three verified cross sections, it can be found that the calculated results are be close to the measured results in cross-sectional shape, and the relative errors ($relative\ error = |calculated\ X - measured\ X|/measured\ X$) of the calculated cross-sectional areas and the calculated wetted perimeters may reach more than 40% at a low water level, but smaller than 5% at a moderate water level (in Figure 8). In river simulation, if the relative error between the constructed riverway model and the actual terrain is not more than 5%, we usually believe that the constructed river model meets the calculational accuracy requirements [23,30]. This means that at a moderate water level, the method presented in this paper is accurate enough to meet the terrain accuracy requirements in actual production. For the exemplified riverway, why the relative errors of the calculated results are large at a low water level is due to the small number of the used longitudinal control lines and the large grid spaces. In this paper, three longitudinal control lines and large spaces are used for the riverway grid generation only to clarify the specific principle of the digital terrain reconstruction method in a concise way. In practical applications, if conditions permit, we should try our best to increase the numbers of the longitudinal control lines and the measured cross sections and choose small grid spaces as far as possible, which can further improve the calculational accuracy.

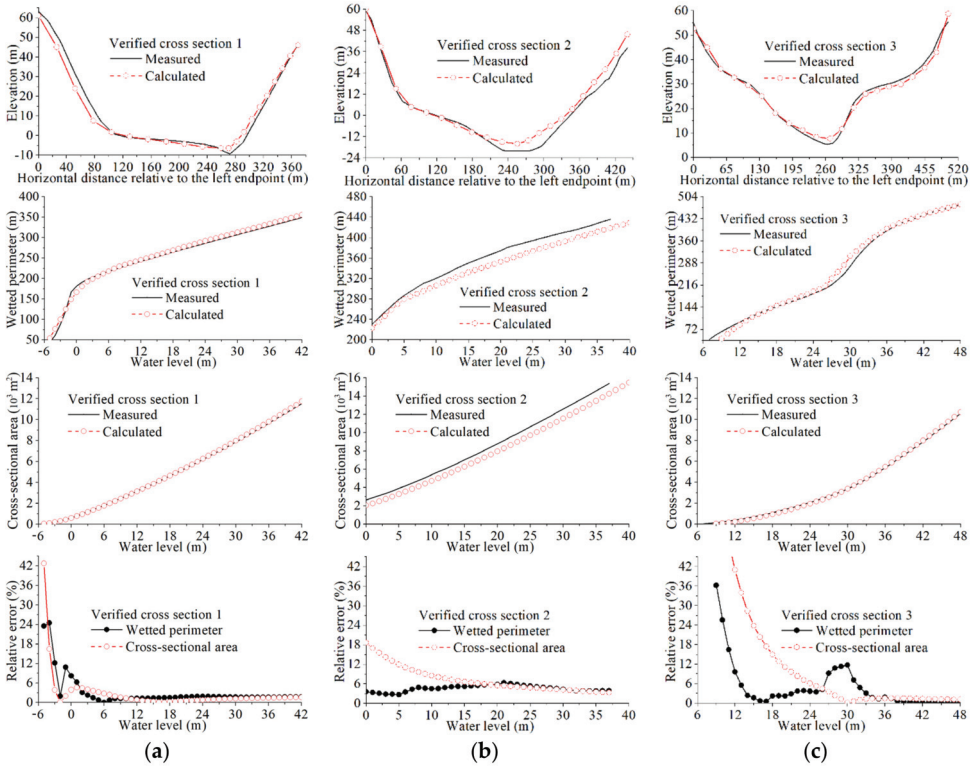


Figure 8. Comparisons between the measured and calculated results related to cross-sectional shape, wetted perimeter and cross-sectional area ((**a**) corresponding to the verified cross section 1; (**b**) corresponding to the verified cross section 2; (**c**) corresponding to the verified cross section 3).

4. Digital Terrain Applications

According to the readable text formats of MIKE21 and SMS, the gridded digital terrain and connection information are output by computer programming (such as Fortran, MATLAB, Python, etc.) to achieve good construction of the data exchange channels and fully exploit the special advantages of various software programs for digital terrain utilization.

4.1. Meshing Digital Terrain for MIKE21

MIKE is a flow simulation module developed by the Danish DHI Company that combines the widely used MIKE11 and MIKE21. MIKE21 is applicable to the argumentation and analysis of macroscopic watershed control engineering, the research on watershed flood dispatching, microscopic flow simulation and other fields. The common grid types include quadrilateral grids. The extension of the MIKE21 quadrilateral grid file is "mesh", and its internal data include the node header line, the node lines, the cell header line, and the cell lines. The node header line is further divided into the entry type with integer form, the entry unit with integer form, the node amount and the character string of the projection type. The entry type is elevation, and its integer form is "100079". The entry unit is the elevation unit, and the integer form "1000" indicates that the elevation values are stored in the z-coordinates and that their units are all meters. The third integer in the node header line is the node amount. The final string "NON-UTM" is the projection type. Each node line represents a node, and the total number of node lines is the same as the node amount in the node header line. The information of each node line includes the node number, x, y, z and boundary code. A boundary code of "0" represents the internal node, "1" represents the water-land boundary, "2" represents the inlet boundary, and "3" represents the outlet boundary. The three numbers in the cell header line indicate the cell amount, the maximum node amount in a single cell, and the cell type code ("25" represents a quadrilateral cell). Each cell line represents a cell, and the total number of cell lines is the same as the cell amount defined in the cell header line. The information in each cell line includes the cell number and the node numbers that constitute the cell.

To use the generated riverway terrain in the hydrodynamic module of MIKE21, it must be saved in the readable grid format of MIKE21 by computer programming. In the specific grid transformation process, the coding rules of the nodes and cells comply with the rules shown in Figure 9 (the bold values in the figure are the cell numbers, and the values at the four corners of the bold values are the node numbers). The node codes and the cell codes are conducted from the riverway inlet to the outlet along the longitudinal grid line, and the transverse output sequence starts from the left bank of the riverway and ends at the right bank. Assume that the number of generated longitudinal grid lines is m, the number of generated transverse grid lines is n, the numbering sequence of longitudinal grid lines is from left to right, and the numbering sequence of transverse grid lines is from the inlet to the outlet (in Figure 9). When each longitudinal grid line is regarded as a row, each transverse grid line is regarded as a column, the row number is indicated by i, the column number is indicated by j, and the cell is indicated by the combination (i, j) of the smallest row number and the smallest column number of its four vertices, then a one-to-one correspondence exists among the cell number, the node numbers constituting the cell, and the transverse and longitudinal grid line numbers. As shown in Figure 10, when the row and column number combination of a certain cell is (i, j) where $i = 1, 2, \ldots, m - 1$ and $j = 1, 2, \ldots, n - 1$, the cell number is calculated as $(i - 1)\cdot(n - 1) + j$. If the nodes constituting the cell are recorded as ①, ②, ③, and ④ along the counterclockwise direction, then the node numbers can be calculated according to formula sets (5). If the cell number is N, the row and column number combination (i, j) of the cell can also be calculated, where i is the minimum integer not less than $N/(n - 1)$, and j equals $N - (i - 1)\cdot(n - 1)$. After i and j are

calculated, the corresponding node numbers can be obtained by substituting i and j into formula sets (5).

$$\begin{cases} : j + (i-1)\cdot n \\ : j + i\cdot n \\ : j + 1 + i\cdot n \\ : j + 1 + (i-1)\cdot n \end{cases} \quad (5)$$

The transverse grid line numbers										
	1		2		3	…	$n-1$		n
The left bank of the riverway									
The longitudinal grid line numbers	1	1		2		3	…	$n-1$		n
	…		**1**		**2**		…		$\boldsymbol{n-1}$	
	2	$n+1$		$n+2$		$n+3$	…	$2n-1$		$2n$
	…		$\boldsymbol{n}$		$\boldsymbol{n+1}$		…		$\boldsymbol{2(n-1)}$	
	3	$2n+1$		$2n+2$		$2n+3$	…	$3n-1$		$3n$
	…	…	…	…	…	…	…	…	…	…
	$m-1$	$(m-2)\cdot n+1$		$(m-2)\cdot n+2$		$(m-2)\cdot n+3$	…	$(m-1)\cdot n-1$		$(m-1)\cdot n$
	…		$\boldsymbol{(m-2)\cdot(n-1)+1}$		$\boldsymbol{(m-2)\cdot(n-1)+2}$		…		$\boldsymbol{(m-1)\cdot(n-1)}$	
	m	$(m-1)\cdot n+1$		$(m-1)\cdot n+2$		$(m-1)\cdot n+3$	…	$m\cdot n-1$		$m\cdot n$
The right bank of the riverway										

Inlet / Outlet

Figure 9. Coding rules of nodes and cells in the MIKE21 quadrilateral grid file (The rows marked "1, 2, … , m" represent longitudinal grid lines; the columns marked "1, 2, … , n" represent longitudinal grid lines; the bold values in the figure are the cell numbers; and the values at the four corners of the bold values are the node numbers.).

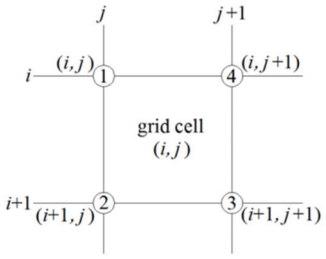


Figure 10. Numbering calculation of nodes constituting a MIKE21 quadrilateral grid cell ("i" and "$i + 1$" represent the serial numbers of the adjacent longitudinal grid lines; "j" and "$j + 1$" represent the serial numbers of the adjacent transverse grid lines; the cell serial number is marked by the combination (i, j); and "①~④" are the node numbers constituting the cell).

When the readable grid file of MIKE21 is generated using the above rules by means of computer programming, it can be imported into the hydrodynamic module of MIKE21 to conduct the numerical simulation. After the grid file is imported into the stated module, the colored terrain map is as shown in Figure 11.

4.2. Meshing Digital Terrain for SMS

The surface water modeling system (referred to as SMS) is a business software program jointly developed by the United States Army Corps of Engineers Hydraulics Laboratory and Brigham Young University. Its quadrilateral grid file extension is "2 dm", and its internal data mainly include a cell line, node line and node strings indicating the open boundaries (inlet and outlet boundaries). The cell lines begin with "E8Q" and are followed by the cell number, the node numbers constituting the cell (a quadrilateral cell in SMS consists of eight nodes—four vertices and the midpoints of four edges) and the material number. Each node line begins with "ND" and is followed by the node number, x and y coordinates, and elevation. The node strings indicating the open boundaries begin with "NS" and are followed by the node numbers of constituting the node strings. The numbering sequence generally starts from the right bank of the riverway and ends with a negative number.

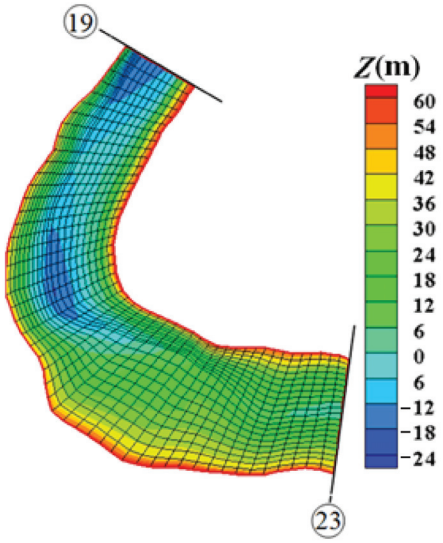


Figure 11. Terrain colored map based on MIKE21.

To use the generated riverway terrain in the two-dimensional flow and sediment transport module of SMS, it must be saved in the readable grid format of SMS by computer programming. However, since a quadrilateral cell in the SMS readable grid format includes eight nodes (four vertices and the midpoints of four edges), only the vertex coordinates of the quadrilateral grid cells are obtained by the abovementioned method. Therefore, before grid conversion, the midpoint coordinates of the corresponding edges should be calculated based on the vertex coordinates of the quadrilateral grid cells. During the specific grid conversion, the coding rules of the nodes and cells can comply with the rules shown in Figure 12 (the bold values in the figure are the cell numbers, and the values around the bold values are the node numbers): First, the vertexes of each quadrilateral grid cell are encoded; second, the midpoints of the transverse edges are encoded; and finally, the midpoints of the longitudinal edges are encoded. The vertex codes and the midpoint codes of the transverse edges are conducted from the riverway inlet to the outlet along the longitudinal grid lines, and the transverse output sequence starts from the left bank and ends at the right bank. The midpoint codes of the longitudinal edges are conducted from the riverway left bank to the right bank along the transverse grid lines, and the longitudinal output sequence starts from the riverway inlet and ends at the outlet. The cell codes between two adjacent longitudinal grid lines start from the riverway inlet and end at the riverway outlet, and the transverse output sequence is from left to right along the transverse grid lines. Assume that the number of generated longitudinal grid lines is m, the number of generated transverse grid lines is n, the numbering sequence of longitudinal grid lines is from left to right, and the numbering sequence of the transverse grid lines is from the riverway inlet to the outlet (in Figure 12). If each longitudinal grid line is regarded as a row, each transverse grid line is regarded as a column, the row number is marked with i, the column number is marked with j, and the cell is marked by the combination (i, j) of the smallest row number and the smallest column number of its four vertices, then a one-to-one correspondence exists among the cell number, the node numbers constituting the cell, and the transverse and longitudinal grid line numbers. As shown in Figure 13, when the row and column number combination of a certain cell is (i, j) where $i = 1, 2, \ldots, m - 1$ and $j = 1, 2, \ldots, n - 1$, then the cell number is calculated as $(i - 1)\cdot(n - 1) + j$. If the nodes constituting the cell are recorded as ①, ②, ③, ④, ⑤, ⑥, ⑦ and ⑧ along the counterclockwise direction, then the numbers can be calculated according to formula sets (6). If a cell number is known as N, the row and column number combination (i, j) of the cell can also be calculated, where i is

the smallest integer not less than $N/(n-1)$ and j equals $N-(i-1)\cdot(n-1)$. After i and j are calculated, the corresponding node numbers can be obtained by substituting i and j into formula sets (6).

$$\begin{cases} : j+(i-1)\cdot n \\ : m\cdot n+j+(i-1)\cdot n \\ : j+i\cdot n \\ : m\cdot n+(m-1)\cdot n+i-1+(j-1)\cdot m \\ : j+1+i\cdot n \\ : m\cdot n+j+1+(i-1)\cdot n \\ : j+1+(i-1)\cdot n \\ : m\cdot n+(m-1)\cdot n+i+(j-1)\cdot m \end{cases} \tag{6}$$

			The transverse grid line numbers								
			1		2		3	...	$n-1$		n
			The left bank of the riverway								
The longitudinal grid line numbers	1	Inlet	1	$(2m-1)\cdot n+1$	2	$(2m-1)\cdot n+m+1$	3	...	$n-1$	$(2m-1)\cdot n+(n-2)\cdot m+1$	n
			$m\cdot n+1$	**1**	$m\cdot n+2$	**2**	$m\cdot n+3$	...	$(m+1)\cdot n-1$	$\boldsymbol{n-1}$	$(m+1)\cdot n$
	2		$n+1$	$(2m-1)\cdot n+2$	$n+2$	$(2m-1)\cdot n+m+2$	$n+3$	...	$2n-1$	$(2m-1)\cdot n+(n-2)\cdot m+2$	$2n$
			$(m+1)\cdot n+1$	$\boldsymbol{n}$	$(m+1)\cdot n+2$	$\boldsymbol{n+1}$	$(m+1)\cdot n+3$	...	$(m+2)\cdot n-1$	$\boldsymbol{2(n-1)}$	$(m+2)\cdot n$
	3		$2n+1$	$(2m-1)\cdot n+3$	$2n+2$	$(2m-1)\cdot n+m+3$	$2n+3$	...	$3n-1$	$(2m-1)\cdot n+(n-2)\cdot m+3$	$3n$
	...		...	...	...	...	...	...	...	...	...
	$m-1$		$(m-2)\cdot n+1$	$(2m-1)\cdot n+m-1$	$(m-2)\cdot n+2$	$(2m-1)\cdot n+2m-1$	$(m-2)\cdot n+3$	...	$(m-1)\cdot n-1$	$(2m-1)\cdot n+(n-1)\cdot m-1$	$(m-1)\cdot n$
			$(2m-2)\cdot n+1$	$\boldsymbol{(m-2)\cdot(n-1)+1}$	$(2m-2)\cdot n+2$	$\boldsymbol{(m-2)\cdot(n-1)+2}$	$(2m-2)\cdot n+3$	...	$(2m-1)\cdot n-1$	$\boldsymbol{(m-1)\cdot(n-1)}$	$(2m-1)\cdot n$
	m	Outlet	$(m-1)\cdot n+1$	$(2m-1)\cdot n+m$	$(m-1)\cdot n+2$	$(2m-1)\cdot n+2m$	$(m-1)\cdot n+3$	...	$m\cdot n-1$	$(2m-1)\cdot n+(n-1)\cdot m$	$m\cdot n$
			The right bank of the riverway								

Figure 12. Coding rules of nodes and cells of the SMS quadrilateral grid file (The rows marked "1, 2, . . . , m" represent longitudinal grid lines; the columns marked "1, 2, . . . , n" represent longitudinal grid lines; the bold values in the figure are the cell numbers; and the values around the bold values are the node numbers).

Using the above rules, after the readable grid file is generated by computer programming, it can be imported into the two-dimensional flow and sediment transport module of SMS to conduct the numerical simulation. After the grid file is imported into the stated module, the three-dimensional riverway grid can be obtained as shown in Figure 14.

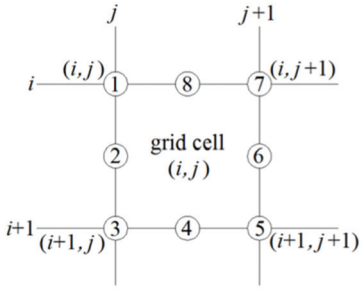


Figure 13. Numbering calculation of nodes constituting an SMS quadrilateral grid cell ("i" and "$i+1$" represent the serial numbers of the adjacent longitudinal grid lines; "j" and "$j+1$" represent the serial numbers of the adjacent transverse grid lines; the cell serial number is marked by the combination (i, j); and "①~⑧" are the node numbers constituting the cell).

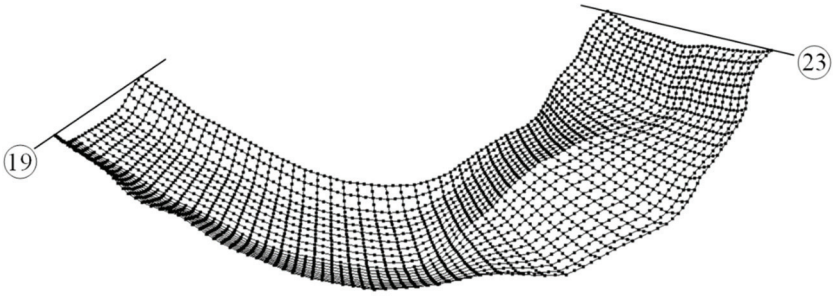


Figure 14. Three-dimensional riverway grids based on SMS.

5. Conclusions

A new method for digital terrain reconstruction using longitudinal control lines and sparse measured cross sections with large spaces is presented. Through interpreting satellite images, remote sensing images or site surveys, the longitudinal control lines such as the river boundaries, the thalweg, the dividing lines of floodplains and main channel, the water edges, etc., can be obtained. Then, the longitudinal control lines are introduced into quadrilateral grid generation as auxiliary lines that can control longitudinal riverway trends and reflect transverse terrain changes. Then, using the equal cross-sectional area principle at a same water level and the weighted distance method, the elevation interpolations for the intersections of the measured cross sections and the longitudinal grid lines are carried out to reasonably fit the measured cross sections. On the above basis, the weighted distance method is used to interpolate all the grid nodes along longitudinal grid lines based on the fitted cross-sectional terrain data. Furthermore, the terrain elevations and connection information at the interpolated grid nodes can be output and integrated by computer programming according to the readable text formats of MIKE21, SMS or other software to achieve good construction of the data exchange channels and fully exploit the special advantages of various software programs for digital terrain visualization and further utilization. Overall, the physical conception of the digital terrain reconstruction method is clear, its implementation is simple and effective, and it is widely applicable to the digital terrain reconstructions in natural riverways using longitudinal control lines and sparse measured cross sections with large spaces.

Author Contributions: Methodology, formal analysis, writing the original draft, Y.P.; review and editing, J.X.; funding acquisition, K.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the National Natural Science Foundation of China (Grant Nos. 51979181, 51539007 and 51279117).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Some or all of the data, models, or 2072 lines of Fortran programming codes used during the study are available from the corresponding author by request.

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

References

1. Bates, P.D.; Marks, K.J.; Horritt, M.S. Optimal use of high-resolution topographic data in flood inundation models. *Hydrol. Process.* **2003**, *17*, 537–557. [CrossRef]
2. Horritt, M.S.; Bates, P.D.; Mattinson, M.J. Effects of mesh resolution and topographic representation in 2D finite volume models of shallow water fluvial flow. *J. Hydrol.* **2006**, *329*, 306–314. [CrossRef]
3. Brasington, J.; Vericat, D.; Rychkov, I. Modeling river bed morphology, roughness, and surface sedimentology using high resolution terrestrial laser scanning. *Water Resour. Res.* **2012**, *48*, 1–18. [CrossRef]

4. Castellarin, A.; Baldassarre, G.D.; Bates, P.D.; Brath, A. Optimal cross-sectional spacing in preissmann scheme 1D hydrodynamic models. *J. Hydraul. Eng.* **2009**, *135*, 96–105. [CrossRef]
5. Lin, W.C.; Chen, S.Y.; Chen, C.T. A new surface interpolation technique for reconstructing 3D objects from serial cross-sections. *Comput. Vis. Graph. Image Process.* **1989**, *48*, 124–143. [CrossRef]
6. Hardy, R.L. Theory and Applications of the Multiquadric-biharmonic Method 20 Years of Discovery 1968–1988. *Comput. Math. Appl.* **1990**, *19*, 163–208. [CrossRef]
7. Beatson, R.K.; Cherrie, J.B.; Mouat, C.T. Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration. *Adv. Comput. Math.* **1999**, *11*, 253–270. [CrossRef]
8. Carr, J.C.; Beatson, P.K.; Cherrie, J.B.; Mitchell, T.J.; Fright, W.R.; Mccallum, B.C.; Evans, T.R. Reconstruction and representation of 3D objects with radial basis functions. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, 28 July–1 August 2001.
9. Zheng, H.; Song, L.; Hou, Y.; Hou, Y. The partition of unity parallel finite element algorithm. *Adv. Comput. Math.* **2015**, *41*, 937–951. [CrossRef]
10. Yokota, R.; Barba, L.A.; Knepley, M.G. PetRBF-A Parallel O(N) algorithm for radial basis function interpolation with gaussians. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 1793–1804. [CrossRef]
11. Yang, M.; Liang, G.T.; Lai, R.X.; Yu, X. Theory and application of digital topography generation based on curved surface interpolation. *J. Hydraul. Eng.* **2007**, *38*, 221–225. (In Chinese)
12. Wagner, B.; Gartner, H.; Santini, S.; Ingensand, H. Cross-sectional interpolation of annual rings within a 3D root model. *Dendrochronologia* **2011**, *29*, 201–210. [CrossRef]
13. Caviedes-Voullieme, D.; Morales-Hernandez, M.; Lopez-Marijuan, I.; Garcia-Navarro, P. Reconstruction of 2D river beds by appropriate interpolation of 1D cross-sectional information for flood simulation. *Environ. Model. Softw.* **2014**, *61*, 206–228. [CrossRef]
14. Lebrenz, H.; Bardossy, A. Geostatistical interpolation by quantile kriging. *Hydrol. Earth Syst. Sci.* **2018**, *23*, 1633–1648. [CrossRef]
15. Weber, D.; Englund, E. Evaluation and comparison of spatial interpolators. *Math. Geol.* **1992**, *24*, 381–391. [CrossRef]
16. Kraus, K. Visulization of the quality of surface and their derivatives. *Photogramm. Eng. Remote Sens.* **1994**, *60*, 457–463.
17. Zimmerma, D.; Pavlik, C.; Ruggles, A.; Armstrong, M.P. An experimental comparison of ordinary and universal kriging and inverse distance weighting. *Math. Geol.* **1999**, *31*, 375–390. [CrossRef]
18. Gichamo, T.Z.; Popescu, I.; Jonoski, A.; Solomatine, D. River cross-section extraction from the ASTER global DEM for flood modeling. *Environ. Model. Softw.* **2012**, *31*, 37–46. [CrossRef]
19. Andes, L.C.; Cox, A.L. Rectilinear inverse distance weighting methodology for bathymetric cross-section interpolation along the mississippi river. *J. Hydrol. Eng.* **2017**, *22*, 04017014. [CrossRef]
20. Aguilar, F.J.; Agüera, F.; Aguilar, M.A.; Carvajal, F. Effects of terrain morphology, sampling density and interpolation methods on grid DEM accuracy. *Photogramm. Eng. Remote Sens.* **2005**, *71*, 805–816. [CrossRef]
21. Chen, Y.; Crowded, R.; Falconer, R.A. Use of measured and interpolated cross-sections in hydraulic river modeling. *Hydraul. Eng. Softw.* **1998**, *19*, 3–12.
22. Florinsky, I.V.; Eilers, R.G.; Manning, G.R.; Fuller, L.G. Prediction of soil properties by digital terrain modeling. *Environ. Model. Softw.* **2002**, *17*, 295–311. [CrossRef]
23. Sun, Z.Q.; Gu, S.; Wang, H.J.; Yang, Z.S.; Bi, N.S. A method for the calculation of erosion and accumulation in the river channel using bathymetric data of river cross section: Based on orthogonal-curvilinear-grid DEM. *Period. Ocean Univ. China* **2018**, *48*, 90–99. (In Chinese)
24. Harada, S.; Li, S.S. Combining remote sensing with physical flow laws to estimate river channel geometry. *River Res. Appl.* **2018**, *34*, 697–708. [CrossRef]
25. Dubey, A.K.; Gupta, P.; Dutta, S.; Kumar, B. Evaluation of satellite-altimetry-derived river stage variation for the braided Brahmaputra River. *Int. J. Remote Sens.* **2014**, *35*, 7815–7827. [CrossRef]
26. Saur, R.; Rathore, V.S. Flashy river channel migration and its impact in the Jiadhal river basin of Eastern Himalaya, Assam, India: A long term assessment (1928–2010). *J. Earth Syst. Sci.* **2022**, *131*, 50. [CrossRef]
27. Khondoker, M.; Siddiquee, M.; Islam, M.A. The challenges of river bathymetry survey using space borne remote sensing in bangladesh. *Atmos. Ocean. Sci.* **2016**, *1*, 7–13.
28. Niroumand-Jadidi, M.; Vitti, A.; Lyzenga, D.R. Multiple optimal depth predictors analysis (MODPA) for river bathymetry: Findings from spectroradiometry, simulations, and satellite imagery. *Remote Sens. Environ.* **2018**, *218*, 132–147. [CrossRef]
29. Mandlburger, G.; Kolle, M.; Nubel, H.; Soergel, U. Bathynet: A deep neural network for water depth mapping from multispectral aerial images. *J. Photogramm. Remote Sens. Geoinf. Sci.* **2021**, *89*, 71–89. [CrossRef]
30. Hua, Z.L.; Wang, H.Y.; Wang, L.; Wang, Y.L. Comparison of different methods for interpolation of topography of discrete rivers. *Adv. Sci. Technol. Water Resour.* **2016**, *36*, 16–19. (In Chinese)

remote sensing

MDPI

Article

Optimized Spatial Gradient Transfer for Hyperspectral-LiDAR Data Classification

Bing Tu [1,*], Yu Zhu [1], Chengle Zhou [1], Siyuan Chen [1] and Antonio Plaza [2]

[1] College of Information and Communication Engineering, Hunan Institute of Science and Technology, Yueyang 414000, China; ZhuYu1103@vip.hnist.edu.cn (Y.Z.); chengle_zhou@vip.hnist.edu.cn (C.Z.); siyuan@hnist.edu.cn (S.C.)

[2] Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politecnica, University of Extremadura, E-10003 Caceres, Spain; aplaza@unex.es

* Correspondence: tubing@hnist.edu.cn

Abstract: The classification accuracy of ground objects is improved due to the combined use of the same scene data collected by different sensors. We propose to fuse the spatial planar distribution and spectral information of the hyperspectral images (HSIs) with the spatial 3D information of the objects captured by light detection and ranging (LiDAR). In this paper, we use the optimized spatial gradient transfer method for data fusion, which can effectively solve the strong heterogeneity of heterogeneous data fusion. The entropy rate superpixel segmentation algorithm over-segments HSI and LiDAR to extract local spatial and elevation information, and a Gaussian density-based regularization strategy normalizes the local spatial and elevation information. Then, the spatial gradient transfer model and l^1-total variation minimization are introduced to realize the fusion of local multi-attribute features of different sources, and fully exploit the complementary information of different features for the description of ground objects. Finally, the fused local spatial features are reconstructed into a guided image, and the guided filtering acts on each dimension of the original HSI, so that the output maintains the complete spectral information and detailed changes of the spatial fusion features. It is worth mentioning that we have carried out two versions of expansion on the basis of the proposed method to improve the joint utilization of multi-source data. Experimental results on two real datasets indicated that the fused features of the proposed method have a better effect on ground object classification than the mainstream stacking or cascade fusion methods.

Keywords: data fusion; gradient transfer; superpixel; hyperspectral image; LiDAR data

Citation: Tu, B.; Zhu, Y.; Zhou, C.; Chen, S.; Plaza, A. Optimize Spatial Gradient Transfer for Hyperspectral-LiDAR Data Classification. *Remote Sens.* **2022**, *14*, 1814. https://doi.org/10.3390/rs14081814

Academic Editors: Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Received: 3 March 2022
Accepted: 5 April 2022
Published: 9 April 2022

1. Introduction

The development of remote sensing sensor technology makes it possible to obtain different types (e.g., hyperspectral image (HSI) and LiDAR) of remote sensing data in the same observation scene, which can capture a full range of identification information of ground coverings in the scene. A hyperspectral image (HSI) can provide rich spectral information for several materials; its high spectral resolution is conducive to distinguishing subtle spectral differences, and thus, making it widely used to identify and classify ground coverings [1–3]. However, the types of ground coverings are often complex, which leads to the phenomenon of the "same spectrum corresponds to multiple ground coverings" [4,5]. And HSI is a spatial flat spectral image degenerated from the real 3D spatial scene; thus, the height information of the observation area is lost. By contrast, LiDAR can obtain the digital surface model (DSM) information of the study area and is not easily restricted by weather or light [6,7]. Therefore, compared with a single data source, effectively combining HSI and LiDAR data and making full use of the complementary advantages of the two will greatly improve the accuracy of ground covering recognition [8,9].

In recent years, many supervised paradigm spectral classifiers have been developed to perform HSI classification tasks, such as the widely used support vector machine

(SVM) [10,11], multinomial logistic regression classifier [12,13] and artificial immune network (AIN) [14]. Although these classifiers can effectively use the spectral information of HSI, they ignore the spatial context information of the pixels. To address this issue, many scholars have proposed a variety of classification methods based on spatial-spectral feature extraction [15–17]. In fact, these spectral-spatial classification methods are dedicated to extracting highly discriminative spatial-spectral features to improve classification accuracy further. For example, Wang et al. [18] design an extremely lightweight, non-deep parallel network (HyperLiteNet) that independently extracts and optimizes diverse and divergent spatial and spectral features. In [19], the adaptive sparse representation algorithm obtains the sparse coefficients of the multi-feature matrix for HSI classification, and these features reflect different kinds of spectral and spatial information. Furthermore, a Global Consistent Graph Convolutional Network(GCGCN) is proposed in [20], which uses graph topology consistent connectivity to explore adaptive global high-order neighbors to capture underlying rich spatial contextual information. The multiway attention mechanism has been successfully applied to HSI analysis due to the inspiration of the attention mechanism of the human visual system [21]. In addition to the above spatial-spectral classification methods, other useful techniques have been encouraged for hyperspectral classification, such as Markov random fields [22,23], collaborative representation [24,25] and edge-preserving filtering [26,27].

As the requirements for the classification of remote sensing scenes continue to increase, it is difficult for the single HSI data to meet the current interpretation task of ground coverings [28–30]. Although HSI data can provide rich diagnostic information (spectral features) for the identification of ground covering, it is limited by its low spatial resolution characteristics, resulting in a performance bottleneck in the classification model. LiDAR is a kind of digital image formed by digital surface model (DSM), which contains richer spatial detail information. In fact, many studies have demonstrated that the interpretation results of the gorund coverings are more accurate and stable by effectively combining the complementary strengths of HSI and LiDAR information [31,32]. For instance, Jia et al. was proposed a multiple feature-based superpixel-level decision fusion (MFSuDF) method for HSIs and LiDAR data classification. The motivation behind the MFSuDF is to considers the magnitude and phase information to obtain discriminative Gabor characteristics of the stacked matrix of HSI and LiDAR. Chen et al. [32] used dual convolutional neural networks (CNNs) to extract features from HSI and LiDAR data and a fully connected (FC) network to fuse the extracted features. These fusion models can extract robust features, but the fusion of HSI and LiDAR data still has many problems that should be explored in depth. Recently, the more popular fusion models adopt the method of features cascade or stacking, which ignores the difference in physical meaning and quantification range of different types of features and cannot encourage complementary information in the description of objects. Furthermore, the stacking mode may lead to information redundancy and Hughes phenomenon, especially in the case of small samples, overfitting may occur.

In the remote sensing community, the superpixel segmentation algorithm as a technique for clustering pixels based on dominant features (such as image color and brightness) has been widely used to extract the local spatial structure information of the pixels [33]. Some new technologies [34,35] that combine the spatial characteristics of superpixels have been proven successful in multi-source remote sensing data fusion tasks and improving the accuracy of ground object interpretation. Furthermore, Jiang et al. [36] introduced a superpixel principal component algorithm (SuperPCA) for HSI classification, which incorporated spatial context information into a superpixel to eliminate the difference of spatial projection between homogeneous regions. In [37], Zhang et al. constructed local-global features by improving the SuperPCA and reconstructed each pixel by exploiting the nearest neighbor pixels in the same superpixel to eliminate noise.

Strong isomerism of features limits the performance of feature fusion classification for heterologous data. The widely used stacking or cascading data fusion methods ignore the problems of different physical meanings, different data forms, and high feature dimensions

describing the same scene with heterogeneous data. Therefore, the fusion method of stacking heterogeneous data cannot effectively achieve complementary information fusion. The basic motivation behind this paper is to use mathematical optimization to fuse the elevation information of single-band LiDAR with the spatial information of hyperspectral images for local feature fusion, which overcomes the high-order nonlinear phenomenon of multi-sensor data space and improves the information fusion performance of multi-dimensional heterogeneous feature discrimination of ground objects.

Specifically, the entropy rate superpixel segmentation algorithm over-segments HSI and LiDAR to extract local spatial and elevation information, and a Gaussian density-based regularization strategy normalizes the local spatial and elevation information. Then, the spatial gradient transfer model and l^1-total variation minimization are introduced to realize the fusion of local multi-attribute features of different sources, and fully exploit the complementary information of different features for the description of ground objects. Finally, the fused local spatial features are reconstructed into a guided image, and the guided filtering acts on each dimension of the original HSI, so that the output maintains the complete spectral information and detailed changes of the spatial fusion features.

In addition, Figure 1a gives the part of the first principal component of the Houston data set. Figure 1b depicts LiDAR data, which contain distinct boundary and objects elevation information. Figure 1c simulates a fusion image obtaining by a stack-based fusion method. Figure 1d is the fusion result of the proposed OSGT algorithm. It can be seen from Figure 1 that the proposed OSGT method can capture more detailed spatial structure information than the stack-based fusion method. Specifically, the main contributions of the proposed OSGT method are summarized as follows.

1. We define homogeneous region fusion between PC and LiDAR data as a mathematical optimization problem and introduce the gradient transfer model to fuse spectral and DSM information from various superpixel blocks for the first time. It is found that the model can alleviate the heterogeneity of different sources of remote sensing data by optimizing the objective function.
2. The l^1-total variation minimization is designed to fuse information between the PC and DSM within each superpixel block to accurately describe the observed details. It is found that the problem of HSI weak boundary affected by the weather can be effectively overcome.
3. The proposed OSGT algorithm can fully extract the complementary features in the homogeneous regions corresponding to HSI and LiDAR to further promote classification of ground coverings competitive methods.

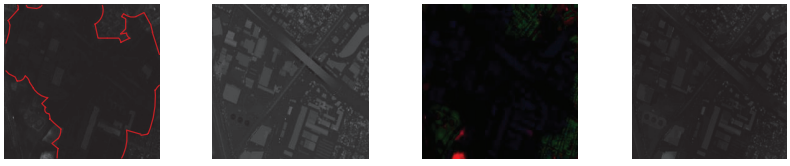


Figure 1. Schematic illustration of image fusion. (**a**) The first PC of the Houston dataset. (**b**) LiDAR data. (**c**) The fusion result of stacking method. (**d**) OSGT-based fusion image.

The rest of this paper is organized as follows. The entropy rate superpixel (ERS) and guided filter (GuF) are reviewed in Section 2, and the proposed OSGF method for HSI and LiDAR data classification is introduced in Section 3. In Section 4, the experimental setup and results are described. Finally, the conclusions of our research are presented in Section 5.

2. Related Work

This section briefly describes some related algorithms, i.e., entropy rate superpixel (ERS), Guided Filtering (GuF). These algorithms play a relevant role in the design of the proposed method.

2.1. Entropy Rate Superpixel (ERS)

Entropy rate superpixel (ERS) [38] is an efficient graph-based over-segmentation method that generates a graph topology of N_s connected subgraphs corresponding to homogeneous superpixels by maximizing the objective function containing an entropy rate term and a balancing term. The ERS method maps the image to a weighted undirected graph $G = (V, E)$, where the node set V and the edge weight E are the pixels of the image and the pairwise similarity given by the similarity matrix, respectively.

Consequently, the segmentation is formulated as a graph division problem, where V is divided into a series of disjoint sets $S = [S_1, S_2, \ldots, S_{N_A}]$, in which the intersection of any two subsets is empty, and the union of all subsets is equal to V. When selecting a subset A of E from $G = (V, E)$ is finished, an undirected graph composed of N_s subgraphs $G^{'} = (V, A)$ is generated. The segmentation problem is formulated as maximizing the following objective function:

$$\max_{A} H(A) + \lambda B(A) \qquad s.t. \quad A \subseteq E \tag{1}$$

where $H(A)$ is the entropy rate of the random walk encouraging uniform and compacting clusters, $B(A)$ represents the balance term controlling clusters with similar sizes, and λ refers to the weight of the constrained entropy rate term and the balance term.

2.2. Guided Filtering

Guided filtering (GuF) [39] is an edge-preserving smoothing filter based on a local linear model. It has been successfully applied to various computer vision tasks, such as image edge smoothing [40], detail enhancement [41], and image fusion denoising [42]. The GuF typically uses a guided image to filter the input image. The output image contains the global features of the input image and the detailed changes of the guided image. The input image and the guided image are denoted as g and I, respectively. The output image is then defined as

$$t = \Phi_{gf}(g, I, r, \xi) \tag{2}$$

where r is the filter window size, and ξ is the normalization parameter. g is a two-dimensional function whose output is linearly related to the guide input:

$$t_i = a_k I_i + b_k, \forall i \in w_k \tag{3}$$

where w_k is a square window with radius r and the linear factors a_k and b_k are fixed values. The gradient of the output image is taken, $\nabla t = a \nabla I$. Therefore, if the guiding image has gradient property, the output image will also encourage the gradient. This is the reason why the GuF can smooth the background and maintain the high quality of the edge. The optimal linear factor a_k and b_k are obtained by minimizing the following cost function:

$$E(a_k, b_k) = \sum_{i \in w_k} \left[(a_k I_k + b_k - g_i)^2 + \xi a_k^2 \right] \tag{4}$$

where ε is the adjustment parameter of a_k. The linear regression analysis method [43] is selected, and the optimal solution expression is written as:

$$a_k = \frac{\frac{1}{|w|} \sum\limits_{i \in w_k} I_i g_i - \mu_k \overline{g_k}}{\sigma^2 + \xi} \tag{5}$$

$$b_k = \overline{g_k} - a_k \mu_k \tag{6}$$

where $|w|$ is the number of pixels in w_k, σ_k^2 and μ_k are the variance and mean of I in w_k, respectively. Similarly, $\overline{g_k}$ is the mean value of g in the window. Considering that pixel i may be contained in many windows, the linear coefficients calculated in different windows are divergent and, thus, the average value of a_k and b_k in the window centered on pixel i is obtained. The output image is then formulated as follows:

$$t_i = \frac{1}{|w|} \sum_{k,i \in w} (a_k I_i + b_k) = \overline{a_i} I_i + \overline{b_i} \tag{7}$$

3. Proposed Approach

In this section, we introduce in detail the architectural steps of the proposed OSGT method for the classification of HSI and LiDAR data. The overall summary of the OSGT method is shown in Figure 2, A pseudo-code of our newly developed OSGT is given in Algorithm 1, and the specific steps are shown below.

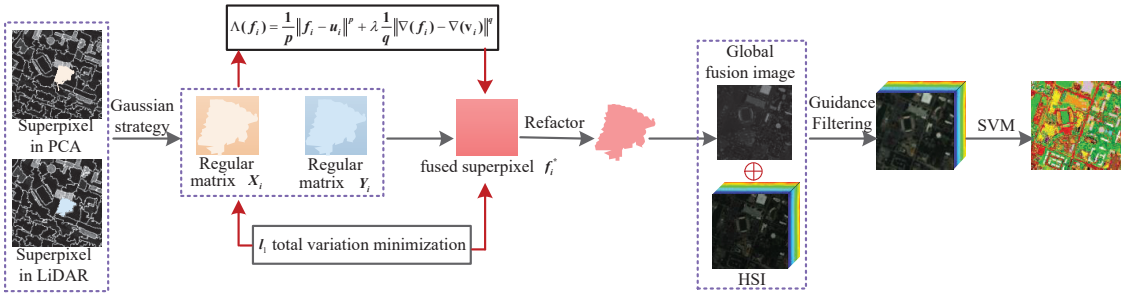


Figure 2. Outline of the proposed OSGT method for hyperspectral and LiDAR data classification.

Algorithm 1: OSGT.

Inputs: the HSI H; LiDAR data L; the number of superpixel N_s; the control parameter λ; the training set T; and test set t;

Outputs: Classification result;

1. **Superpixel Oversegmentation**
   Obtain H_s and L_s based on PCA for H and L, by Equation (1)
   For i = 1:N_s
      Regularization strategy transforms S_i and I_i into X_i and Y_i
   End
2. **optimize spatial gradient transfer algorithm**
   For i = 1:N_s
      Determine y_i^*, by Equations (8)–(12)
      Obtain the fused superpixel blocks $f_i^* = y_i^* + v_i$
      Reconstruct the fused superpixel blocks
   End for
   Generate the fused image F
3. **Classification**
   Use F as the guided image to filter H, by Equation (13)
   Apply SVM to classify

3.1. Oversegmentation

The hyperspectral cube $H \in \mathbb{R}^{M \times N \times B}$ is composed of hundreds of continuous spectral bands. M, N, and B are the numbers of image rows, columns, and spectral channels, respectively. We have an observed 3D hyperspectral dataset in the 2D matrix form,

$H_x \in^{D\times B} (D = M \times N)$, in which each column represents a pixel vector. Similarly, let $L \in \mathbb{R}^{M\times N}$ denote the LiDAR data.

The superpixel segmentation algorithm divides the target image into many disjointed regions. The samples in each region have the same or similar texture, color, and brightness. Assuming that the number of superpixels is N_s, the oversegmented images of the first principal component of H and L are $H_s = \{S_1, S_2, \ldots, S_{N_s}\}$ and $L_s = \{I_1, I_2, \ldots, I_{N_s}\}$, where S_i and I_i represent the superpixel blocks of H and L, respectively.

In order to alleviate the problem of the weak boundary of the super pixel, and the negative influence of the weak boundary on the edge gradient of the super pixel. As shown in Figure 3, a local space mean regularization strategy based on Gaussian density is designed. Specifically, the Gaussian kernel function calculates the samples density of S_i and I_i to describe the information between adjacent samples, and then averages the Gaussian density to fill the irregular S_i and I_i into regular matrix (i.e., X_i and Y_i, $i \in \{1, 2, \ldots, N_s\}$), which not only maintains the spatial information of the superpixels but also avoids excessive edge gradient.

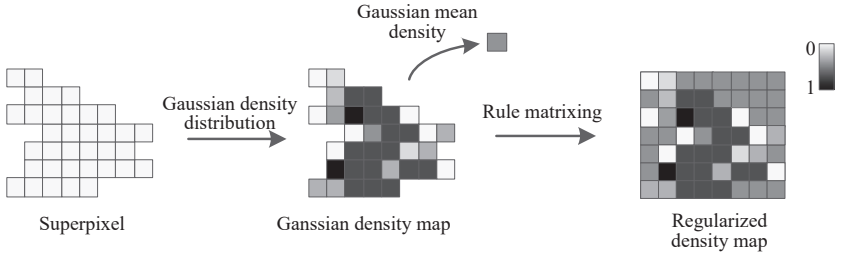


Figure 3. A local spatial mean regularization strategy based on Gaussian density.

3.2. The Proposed OSGT Method

(1) Superpixel-guided gradient transfer fusion: The goal that the superpixel block X_i of HSI and the superpixel block Y_i of LiDAR fuse is to generate a fusion image that contains both spectral information and elevation features. X_i, Y_i and the fusion result can be regarded as grayscale images with a scale of $m \times n$, and their column vector forms are represented by u_i, v_i,$f_i \in \mathbb{R}^{mn\times 1}$, respectively.

HSI contains dense spectral information and scene detail information, and its high spectral resolution is conducive to distinguish the difference of different materials, which restricts the fusion result f_i should have similar pixel intensity to u_i. For the empirical error measured by l^p norm should be as small as possible.

$$\Lambda_1(f_i) = \frac{1}{p}\|f_i - u_i\|^p \tag{8}$$

The spatial dimension data of HSI is actually a 2-D image, but fusion image showing the 3-D spatial information of the observation area is necessary based on the importance of visual perception. The gray value of each point in the LiDAR image reflects the elevation information of the point and hence, we design fusion image f_i to maintain similar pixel gradients instead of intensity to v_i. For this, the error that is measured by l^q norm must be as small as possible and is as follows:

$$\Lambda_2(f_i) = \frac{1}{q}\|\nabla(f_i) - \nabla(v_i)\|^q \tag{9}$$

We define the fusion problem of superpixel blocks X_i and Y_i as minimizing the following objective function:

$$\Lambda(f_i) = \tfrac{1}{p}\|f_i - u_i\|^p + \lambda\tfrac{1}{q}\|\nabla(f_i) - \nabla(v_i)\|^q \tag{10}$$

Here, the first term of the objective function is the data fidelity term, which indicates that f_i should have the same pixel intensity as u_i. The second term is the regularization term, which guarantees the same gradient information of f_i and v_i. λ is the control parameter that constrains the data fidelity term and regularization term. The objective function transfers the gradient information or elevation information of Y_i to the corresponding position in X_i.

(2) Total variation minimization: When the relationship between the fusion image and the constraint target is Gaussian, the l^2 norm is appropriate. However, We expect that the fusion result will be encouraged to retain more features of X_i, as HSI exhibits texture features and spatial features, etc. besides spectral information. Therefore, most entries of f_i and u_i should be the identical. Only several entries are relatively large due to the gradient transfer of v_i, so here ($p = 1$) is the appropriate choice in this paper. In contrast, enhancing the sparsity of LiDAR image gradients can rely on minimizing LiDAR image l^0, i.e., ($q = 0$). However, the l^0 norm is NP-hard; thus, we replace l^0 with l^1, implying that $q = 1$.

Let $y_i = f_i - v_i$, the optimization problem (10) can be rewritten as:

$$y_i^* = \arg\min_{y_i} \left\{ \sum_{j=1}^{mn} \left| y_{ij} - \left(u_{ij} - v_{ij} \right) \right| + \lambda J(y_i) \right\} \tag{11}$$

$$J(y_i) = \sum_{j=1}^{mn} \left| \nabla_{ij} y_i \right| = \sum_{j=1}^{mn} \sqrt{\left(\nabla_{ij}^h y_i \right)^2 + \left(\nabla_{ij}^v y_i \right)^2} \tag{12}$$

where $|a| = \sqrt{a_1^2 + a_2^2}$ for every $a = (a_1, a_2) \in \mathbb{R}^2$. ∇_{ij}^h and ∇_{ij}^v represent the horizontal and vertical gradients of pixel j, respectively. The objective function in Equation (11) is solved directly using the proposed algorithm in [44]. y^* is obtained by optimizing Equation (11) using the technique of l^1-TV minimization; thus, the target fusion outcome f_i^* is decided by $f_i^* = y_i^* + v_i$.

(3) Compute the global optimal solution: N_s hyperspectral image superpixel blocks X_i and LiDAR data superpixel blocks Y_i have been obtained in Section 3.1 . The total variable minimization method optimizes the objective function to fuse superpixel pairs. We denote by $\left\{ f_1^*, f_2^*, \ldots, f_i^*, \ldots, f_{N_s}^* \right\}$ the column-vector form of the fusion result set of N_s superpixel pairs, and the regular matrix form of the fusion result is expressed as $\{r_1, r_2, \ldots, r_i, \ldots, r_{N_s}\}$. We perform superpixel refactor technology. Specifically, the position information of each pixel of S_i and I_i is used to select the pixel of the corresponding position in r_i, and then an irregular superpixel block with the same size as the superpixel S_i and I_i is obtained. Finally, the N_s inverted superpixel blocks are combined into a global fusion image F, where $F \in \mathbb{R}^{M \times N}$.

3.3. Classification for HSI and LiDAR Data

One of the important factors affecting the filtering result is the guiding image, and the gradient of the output image obtained by guiding filtering is completely determined by the gradient of the guiding image.

In Section 3.2, the proposed method fuses HSI and LiDAR into a single-band image F. To some extent, it can be considered that the proposed method transfers the elevation information of the LiDAR data to the corresponding position of the HSI. Therefore, the fused image looks like the first principal component of HSI, but supplements the spatial detail information and cloud occlusion information to make the boundary contour of the object of interest more complete.

We choose the fusion image F as the guiding image, and the original hyperspectral image H as the guiding filter input. Specifically, given the guiding filter window radius r, and the filter ambiguity ξ, we can obtain the following filtering equation:

$$FH = GF_{r,\xi}(H, F) \tag{13}$$

The filtered output can preserve the overall features of the input image and the detailed changes of the guided image through the adjustment of related parameters. It is worth mentioning that the samples in Figure 4 are closer to each other than the samples in Figure 4a, which indicates that the samples in Figure 4b have a higher quality. The structure transfer characteristics of the guided filtering can eliminate edge blocking effects and enhance the ability of feature expression. The filtered features are passed through the SVM classifier to obtain the final classification result.

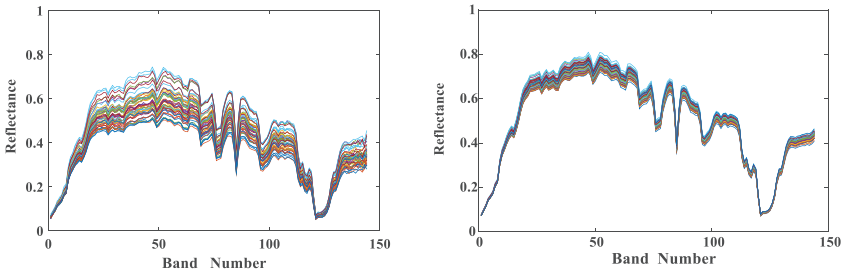


Figure 4. Spectral characteristics (**a**) before and (**b**) after Guided filtering. We take class Railway in Houston dataset for example.

3.4. Extension Method

In this section, the two extended methods we propose are implemented from the perspectives of band fusion to reduce data dimensionality and multi-branch to enrich detailed information, respectively.

(1) We propose an optimized spatial gradient fusion algorithm based on band grouping cooperation aiming to reduce dimensionality while maintaining the physical properties of the data. Since the adjacent bands of hyperspectral image are redundant and highly correlated, the fusion operation can reduce dimensionality and reduce image noise. Specifically, BG-OSGT does not change the main algorithm structure of OSGT. It divides and fuses the filter result graph obtained by the OSGT algorithm instead of directly using SVM for classification. In Section 3.3, the filtering feature map is determined, and in this section we divide it into K adjacent band subsets in the spectral dimension. The kth ($k \in (1, \ldots, K)$) group is defined as follows:

$$\mathrm{P}_k = \begin{cases} (x_k, \ldots, x_{k+\lfloor B/K \rfloor}), & \text{if} \quad k + \lfloor B/K \rfloor \leq B \\ (x_k, \ldots, x_B), & \text{otherwise} \end{cases} \tag{14}$$

where $\mathrm{x} = (x_1, \ldots, x_B) \in \mathbb{R}^{B \times D}$ denotes the filtering feature map containing B feature vectors and D pixels, and then $\lfloor B/K \rfloor$ represents an integer not greater than B/K. Then, the adjacent bands in the kth group are fused by the mean value strategy, that is, the calculation formula of the fusion feature R_k of the kth group is:

$$R_k = \frac{\sum_{i=1}^{N_k} P_k^i}{N_k} \tag{15}$$

where P_k^i is the ith band in the kth band grouping and N_k is the total number of bands in the kth band grouping.

By taking advantage of the each grouping feature, the decision fusion strategy can effectively increase the classification accuracy. Specifically, we fused the label information of each test pixel predicted by different groups. The final classification map is determined by

$$F_c = \underset{c=1,\ldots,G}{\arg\max} \sum_{i=1}^{K} \chi(l_i = c) \tag{16}$$

where F_c is the class label from one of the G possible classes for the test pixel, χ represents the indicator function. Algorithm 2 describes the overall process of the method.

Algorithm 2: BG-OSGT.

Inputs: H; L; N_s; λ; T; and t;

Outputs: Classification result;

1. **Superpixel Oversegmentation**
   Obtain H_s and L_s based on PCA for H and L, by Equation (1)
   For i = 1:N_s
      Regularization strategy transforms S_i and I_i into X_i and Y_i
   End
2. **optimize spatial gradient transfer algorithm**
   For i = 1:N_s
      Determine y_i^*, by Equations (8)–(12)
      Obtain the fused superpixel blocks $f_i^* = y_i^* + v_i$
      Reconstruct the fused superpixel blocks
   End for
   Generate the fused image F
3. **Classification**
   Use F as the guided image to filter H, by Equation (13)
   Apply band grouping strategy to the filtered result
   Multi-branch classification and decision fusion by using SVM

(2) The first principal component of hyperspectral image contain most of the main information, the OSGT algorithm fuses the first principal component of hyperspectral image with LiDAR data. However, if only the first principal component is encouraged, some details may be lost. As shown in Figure 5, there is still information available in the second and third principal components. Therefore, the multi-branch optimize spatial gradient transfer (MOSGT) decision fusion framework is proposed, which aims to enrich image details and corner pixels. Specifically, MOSGT uses the OSGT algorithm to fuse the first three principal components of hyperspectral image with LiDAR data to generate three fused images. Then, the fused feature maps are used as guide images to filter the original hyperspectral image to obtain filtered feature maps, which can make full use of the complementary information between different guide images. In this section, we still use the majority voting decision strategy due to its insensitivity to inaccurate estimates of posterior probabilities.

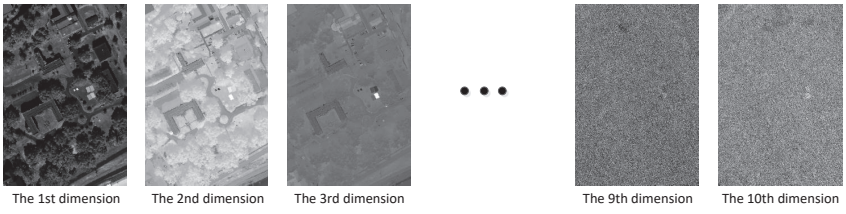


Figure 5. The first 10 principal component images of the MUUFL Gulfport dataset are based on PCA algorithm.

Algorithm 3: MOSGT.

Inputs: H; L; N_s; λ; T; and t;

Outputs: Classification result;

1. Extract the first three principal components M_i^{pc} of H, where $i = 1, 2, 3$
2. Oversegmented M_i^{pc} and L by using ERS method, and then generate oversegmented maps S_i^{pc} and L_s
3. Apply Gaussian regularization strategy to S_i^{pc} and L_s
4. Fusion of S_i^{pc} and L_s according to (8)–(12)
5. Obtain the fused superpixel blocks $f_i^* = y_i^* + v_i$
6. Reconstruct the fused superpixel blocks
7. Generate the fused image set F_i
8. Use F_i as the guided images to filter H, by Equation (13)
9. Classify filtering feature images and decision fusion strategy.

4. Experimental Results

4.1. Datasets

(1) Houston Dataset: The University of Houston image is over the University of Houston campus and surrounding area [9]. It is composed of HSI and LiDAR data, both of which have a spatial dimension of 349 × 1905 and spatial resolution is 2.5 m per pixel. The HSI used in the experiments contains 144 bands, and the wavelength ranges from 380 to 1050 nm. Figure 6 illustrates the false-color composite of the University of Houston image, a grayscale image of the LiDAR data, and the corresponding reference data—there are 15 different classes. The exact numbers of samples for each class are reported in Table 1.

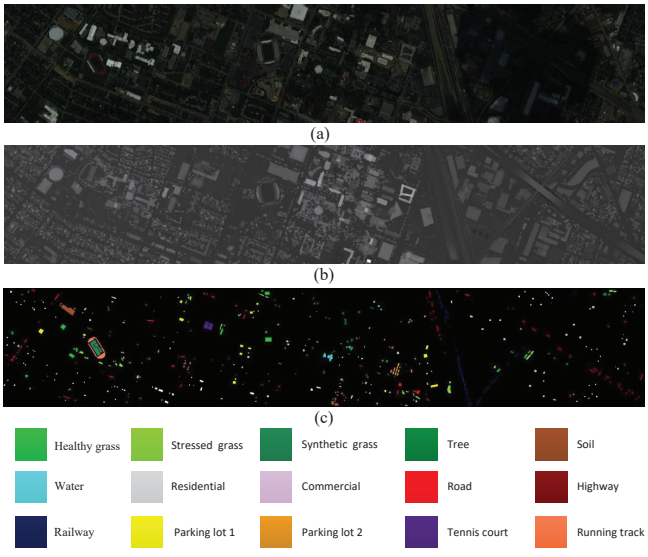
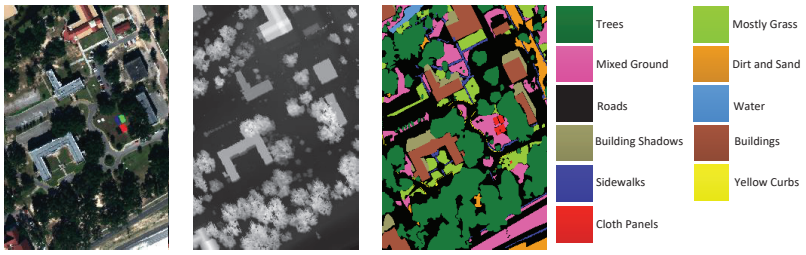


Figure 6. Visualization of the Houston data. (**a**) Pseudo-color image for the hyperspectral data. (**b**) Grayscale image for the LiDAR data. (**c**) Ground truth.

(2) MUUFL Gulfport Dataset: The MUUFL Gulfpor image is over the University of Southern Mississippi Gulfport Campus [45,46]. The HSI has a spatial dimension of 325 × 337 and 72 spectral bands. After discarding 8 bands contaminated by noise, the image contains 64 bands. Furthermore, considering the invalid area of the scene, the original hyperspectral is cropped to 325 × 220 × 64 as the new data set. The false-color composite of MUUFL Gulfport, a grayscale image of the LiDAR data, and the corresponding

reference data are shown in Figure 7. The nine land-cover classes are described in detail in Table 1.



(a) (b) (c)

Figure 7. Visualization of the MUUFL Gulfport. (**a**) Pseudo-color image for the hyperspectral data. (**b**) Grayscale image for the LiDAR data. (**c**) Ground truth.

Table 1. Different numbers of training and testing samples for fifteen classes in the Houston and eleven classes in the MUUFL Gulfport.

Houston				MUUFL Gulfport			
Class	**Land-Cover Type**	**Training**	**Test**	**Class**	**Land-Cover Type**	**Training**	**Test**
C1	Healthy grass	10	1241	C1	Trees	10	23,236
C2	Stressed grass	10	1244	C2	Mostly Grass	10	4260
C3	Synthetic grass	10	687	C3	Mixed Ground	10	6872
C4	Tree	10	1234	C4	Dirt and Sand	10	1816
C5	Soil	10	1232	C5	Roads	10	6677
C6	Water	10	315	C6	Water	10	456
C7	Residential	10	1258	C7	Building Shadows	10	2223
C8	Commercial	10	1234	C8	Buildings	10	6230
C9	Road	10	1242	C9	Sidewalks	10	1375
C10	Highway	10	1217	C10	Yellow Curbs	10	173
C11	Railway	10	1225	C11	Cloth Panels	10	259
C12	Parking lot 1	10	1223				
C13	Parking lot 2	10	459				
C14	Tennis court	10	418				
C15	Running track	10	650				
Total		150	14,879			110	53,577

4.2. Quality Indexes

In order to objectively evaluate the performance of the proposed methods (i.e., the OSGT, BG-OSGT, and MOSGT method), the experiments adopt three objective indicators, i.e., overall accuracy (OA), average accuracy (AA), and Kappa coefficient. OA refers to the probability that the classification result is consistent with the ground truth. AA considers the imbalance of the number of samples in different classes. Kappa represents the consistency between the classification results and the true classes of ground objects—the greater its value, the more accurate the classification result. To eliminate the influence of randomness, the results of all quantitative indicators are averages of ten results.

4.3. Analysis of Parameters Influence

(1) Effect of number of superpixels: In this section, the effect of the number of superpixels on the performance of the proposed OSGT method is evaluated on the Houston and MUUFL Gulfport dataset. As shown in Figure 8, it can be seen that the performance of the proposed OSGT method decreases significantly when the number of superpixels is less than 500 or 600. However, when the number of superpixels is higher than 500 or 600, the classification accuracy slowly decreases. The primary reason is that the large homogeneous region (small number of superpixels) causes the oversegmented map to contain many

boundary superpixels that need to be further segmented. And a smaller homogeneous region (large number of superpixels) leads to poor discrimination of features in the regions. Furthermore, a small number of superpixels can reduce the computational cost. Therefore, the number of superpixels is fixed to 500 for the Houston data set and 650 for the MUUFL Gulfport dataset in this work.

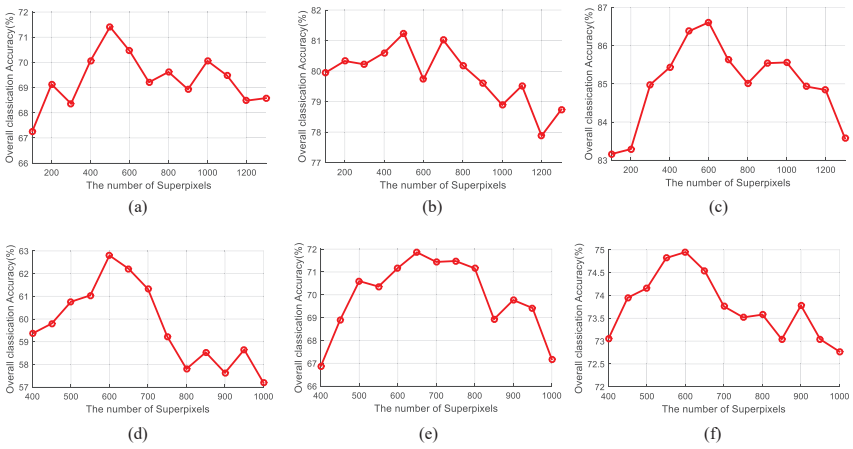


Figure 8. Effect of the number of superpixels on the overall classification accuracy (%) of OSGT method for Houston (**a–c**) and MUUFL Gulfport (**d–f**) datasets. Different numbers of training samples determine the results of each column. Specifically, the first to third columns are the classification accuracy when the number of training samples is 5, 10, and 15 per class, respectively.

(2) Effect of window radius and ambiguity of the GuF: The influence of two parameters, i.e., the window radius r and ambiguity ξ of the guided filtering, are analyzed on the above datasets. Figure 9 illustrates the OA versus r and ξ on different datasets; OA decreases significantly as the window radius r increases. When r and ξ are very small, useful detailed information and corner pixels can be determined. For the Houston dataset, the proposed OSGT method achieves the highest OA when r is set to 2 and ξ is equal to 0.2. For the MUUFL Gulfport dataset, when $r = 1$ and $\xi = 2.5 \times 10^{-3}$, the OSGT method obtains satisfactory classification accuracy.

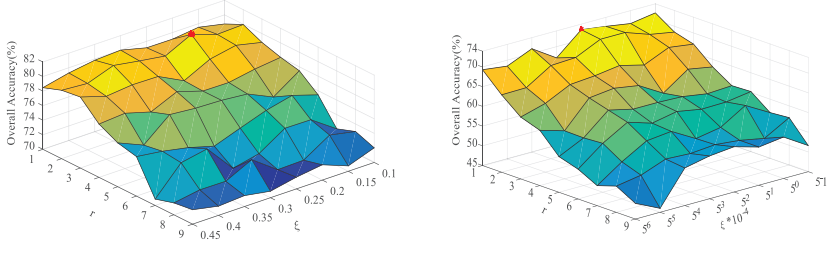


Figure 9. Effect of r and ξ to the performance of classification for different datasets. (**a**) Houston dataset. (**b**) MUUFL Gulfport dataset.

(3) Effect of free parameter: The free parameter λ that controls the data fidelity and regularization terms of the objective function impacts the performance of the proposed OSGT method. Figure 10 illustrates the visualized fusion results and the quantitative index OA under different free parameters for the Houston dataset. As λ increases, the fusion image contains the more abundant elevation information of LiDAR data. However, when

the λ is too high, a small amount of detailed information disappears in the fused image because they only belong to HSI. Our goal is to retain more information existing in HSI, so that the fusion image still resembles HSI. When λ is set to approximately 5, the fusion result retains the small-scale details of the edge of HSI and adds the elevation feature of the ground object. Similarly, Figure 11 reflects that the parameter λ can balance the detailed appearance information and elevation features of the MUUFL Gulfport dataset. When λ is equal to approximately 6, the fusion result is satisfactory.

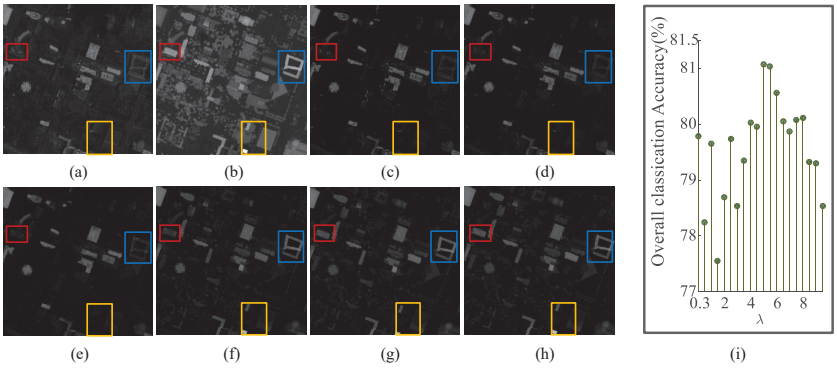


Figure 10. Visualized fusion results and quantitative indicator under different free parameters for Houston dataset. (**a**) The first PC image, (**b**) LiDAR data, (**c**–**h**) Fusion result when $\lambda = 0.1, 0.5, 1, 5, 10, 50$, respectively. (**i**) Overall Accuracy.

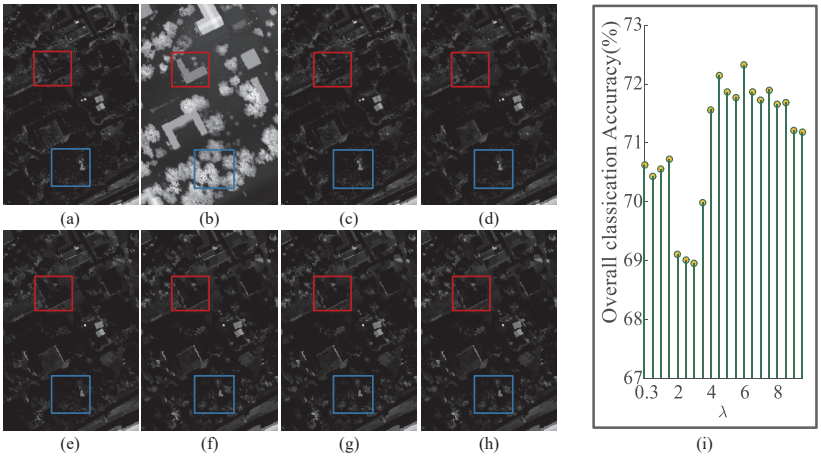


Figure 11. Visualized fusion results and quantitative indicator under different free parameters for MUUFL Gulfport dataset. (**a**) The first PC image, (**b**) LiDAR data, (**c**–**h**) Fusion result when $\lambda = 0.1, 0.5, 1, 5, 10, 50$, respectively. (**i**) Overall Accuracy.

4.4. Analysis of Auxiliary between HSI and LiDAR Data

In this section, the auxiliary effect of LiDAR data on HSI is analyzed on the Huston and MUUFL Gulfport datasets. In this experiment, the numbers of training and test samples are selected to the same as those presented in Table 1. SVM-HSI indicates that SVM classifies the original HSI. G-PCA and G-LiDAR indicate, respectively, that the first PC and the LiDAR data are used as a guide image to filter the original HSI. G-PL represents that LiDAR data is stacked as a band of HSI to form a new dataset, and then the new dataset is then filtered using the first PCs as a guide image. For ensuring the experiment's validity, a spatial mean strategy based on Gaussian density is used for the guide images of the comparison

methods. The super-segmented map of the guide image is subjected to Gaussian density mean filtering in the homogenous region. The experiment is respectively performed on the Houston and MUUFL Gulfport datasets to verify the classification performance of the proposed OSGT method.

As shown in Figure 12, the SVM-HSI leads to an unsatisfactory classification accuracy (OA = 76.32%), indicating that the original HSI contains fuzzy boundary information and noise. Therefore, the classification performance of HSI without any preprocessing must often be improved. Furthermore, the classification accuracy of G-PCA and L-PCA are similar. This phenomenon indicates that the small-scale detail information of the PCA and the elevation attributes of the ground features encouraged by LiDAR can be transferred to the output of the filter as the structure of the guide image. Although their classification performance is similar, the features used to guide filtering differ. G-PL does not significantly improve classification performance because although HSI and LiDAR data are combined into cascaded data, the stacking of two different information expression forms ignores the feature heterogeneity. The proposed OSGT method in this paper fuses multi-source data from the perspective of mathematical optimization, causing the fusion result to contain both appearance detail information and elevates the features of the ground objects so that the guide image structure information is closer to the ground truth value.

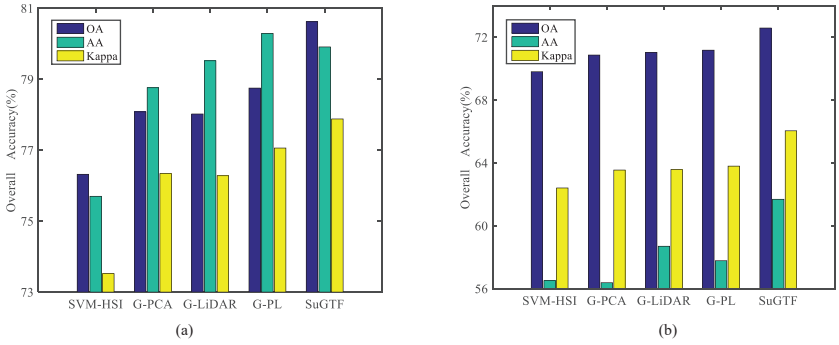


Figure 12. Analyze the auxiliary effect of LiDAR data on HSI for (**a**) Houston dataset and (**b**) MUUFL Gulfport.

4.5. Effect of Filtering Method

In this section, we analyze the impact of different filtering methods on the performance of the proposed OSGT method, by comparing five widely used filtering methods: Gabor filtering (GaF) [47], mean filtering (MF) [48], recursive filtering (RF) [49], bilateral filtering (BF) [50] and guided filtering (GuF).

Figure 13 reports the classification accuracy of the above filtering methods. The cascaded data combined with HSI and LiDAR data are used as the filtering input to test the performance of these filtering methods in terms of extraction of structural information for multi-source data. The relevant parameters adopt the default parameter settings; the GuF parameters are the same as the parameters of the proposed OSGT method. Additionally, as shown in Figures 14 and 15, it can be seen that the GuF method pays more attention to edge detail information and effectively retains the overall spatial features of the input image. Although the accuracy of the MF is slightly higher than GuF on the MUUFL Gulfport data set, its filtering performance of the MF on the Houston data set is significantly worse.

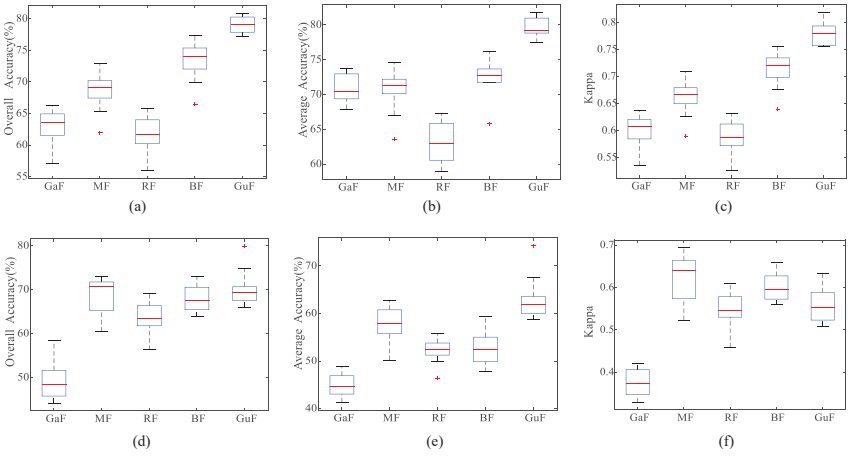


Figure 13. Classification accuracy (i.e., OA, AA, and Kappa) of the proposed OSGT method using different filtering methods. (**a**) OA, (**b**) AA and (**c**) Kappa for Houston, (**d**) OA, (**e**) AA and (**f**) Kappa for MUUFL Gulfport dataset.

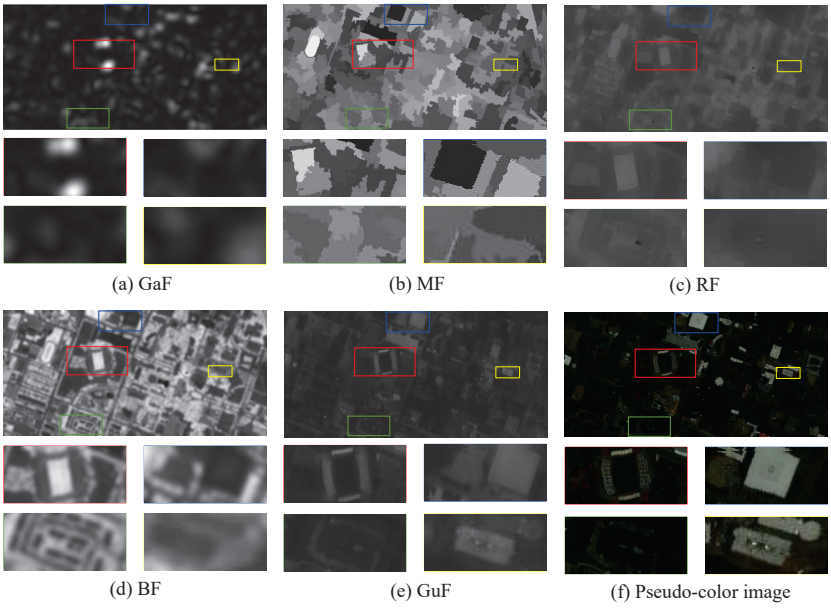


Figure 14. Visualization results of the proposed OSGT method with different filtering methods (i.e., (**a**) Gabor filtering (GaF), (**b**) mean filtering (MF), (**c**) recursive filtering (RF), (**d**) bilateral filtering (BF) and (**e**) guided filtering (GuF).) and (**f**) pseudo-color image on the Houston dataset.

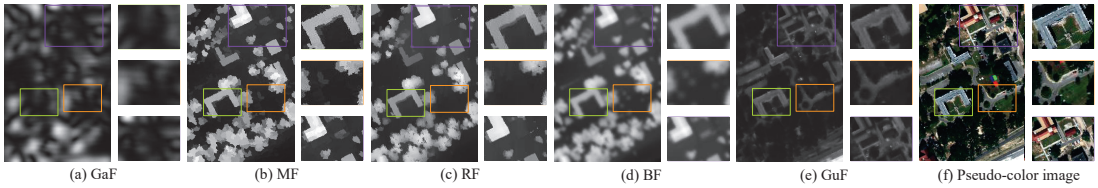


Figure 15. Visualization results of the proposed OSGT method with different filtering methods (i.e., (**a**) Gabor filtering (GaF), (**b**) mean filtering (MF), (**c**) recursive filtering (RF), (**d**) bilateral filtering (BF) and (**e**) guided filtering (GuF).) and (**f**) pseudo-color image on the MUUFL Gulfport dataset.

4.6. Effect of Local Features and Global Features

In this section, the influence of global and local features on image fusion is analyzed in Table 2. The operation of LiDAR images without superpixel processing directly as a guide map is denoted as NSL and the operation of the PCs images without superpixel processing directly as a guide map is denoted as NSP. PCL-GTF and PC^{3}L-GTF indicate that the first PC and the first three PCs, respectively, are fused with the LiDAR data by global gradient transfer. It can be observed from Table 2 that the proposed OSGT method achieves the highest classification accuracy in terms of OA, AA, and Kappa. It is advantageous to fuse the PCs image and LiDAR data in a homogeneous region because the local explicit correlation of superpixels and the homogeneous regions of HSI can be used as the spatial structure information of spatial-spectral classification, enriching the fusion results.

Table 2. Classification accuracy (in %) of Houston and MUUFL Gulfport with no superpixels, global feature fusion and local feature fusion methods.

Houston Data Set					
metrics	NSL	NSP	PCL-GTF	PC^{3}L-GTF	OSGT
OA(%)	78.12	76.39	79.83	79.55	81.02
AA(%)	78.5	76.66	80.66	80.25	81.09
Kappa	0.76	0.75	0.78	0.78	0.79
MUUFL Gulfport Data Set					
metrics	NSL	NSP	PCL-GTF	PC^{3}L-GTF	OSGT
OA(%)	68.68	68.85	70.82	71.12	72.59
AA(%)	52.71	56.58	56.06	56.54	68.69
Kappa	0.61	0.61	0.63	0.64	0.68

4.7. Comparisons with Other Approaches

A series of experimental verifications are conducted on the Houston and MUUFL Gulfport dataset to verify the effectiveness of the proposed OSGT method. The proposed OSGT method is compared with seven other methods. The specific details of the comparison methods are as follows:

1. SVM:The SVM classifier is applied to stacked HSI and LiDAR data, i.e., H.
2. SuperPCA: The SVM classifier is applied to H.
3. CNN: convolutional neural network [51] for HSI and LiDAR data.
4. ERS: SVM classifier is applied to H, and ERS guides the first three PCs to use the spatial mean strategy based on Gaussian density.
5. NG-OSGT: SVM directly classifies the fusion image obtained by the proposed OSGT method.
6. BG-OSGT: Fusion image band grouping cooperation.
7. MOSGT: Multi-branch decision fusion of the first three PCs and LiDAR data.

Specifically, the SVM parameters are set through five layers of cross-validation, and the parameters of SuperPCA and CNN in the comparison method are the default parameters in the corresponding paper.

The first experiment is conducted on the Houston dataset. Table 1 illustrates the number of training samples and testing samples. The classification performance obtained by different methods is shown in Table 3, and the top results for each class are highlighted in bold typeface. The visual classification maps associated with the corresponding OA of different methods are depicted in Figure 16. As shown in Table 3, SVM just considers the spectral information, so the value of OA is only 78.80%. The problem of cascaded data that does not consider data heterogeneity is most evident in SuperPCA. The heterogeneity of data increases the prominence of the implicit irrelevance pixels within the superpixel, limiting classification performance. Moreover, ERS alleviates the problem of implicit irrelevance using the spatial mean strategy based on Gaussian density. However, the heterogeneity of multi-source data is still the most important influencing factor. The deep learning method represented by CNN has poor algorithm performance under small-sample conditions. NG-OSGT does not illustrate excellent classification accuracy because the fusion result has only one band, and the rich spectral information of HSI is lost. Our purpose is to supplement HSI information with the elevation attribute of LiDAR data as an auxiliary item, rather than abandon the spatial-spectral features of HSI. Consequently, OSGT, BG-OSGT, and MOSGT improve the accuracy of the classifier for ground objects identification.

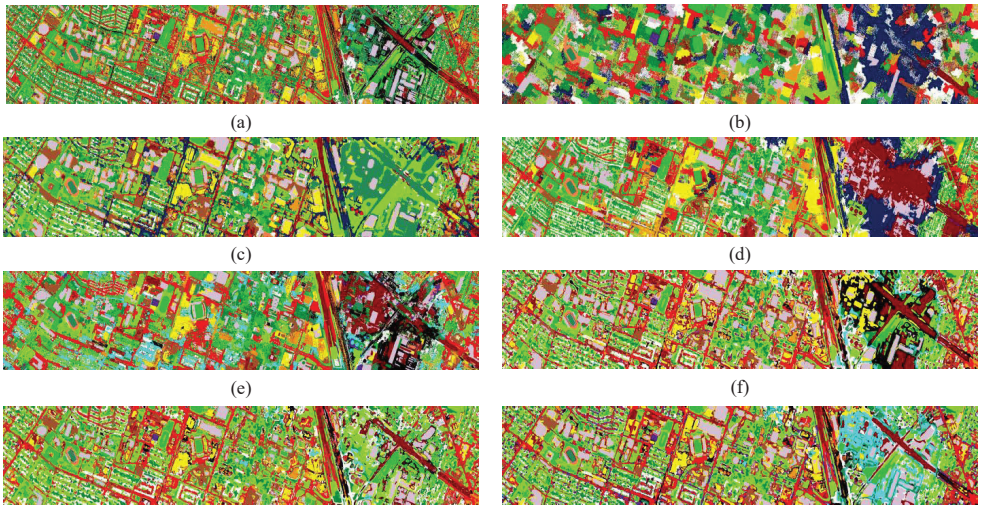


Figure 16. Houston dataset: classification maps obtained by: (**a**) SVM, (**b**) SuperPCA, (**c**) CNN, (**d**) ERS, (**e**) NG-OSGT, (**f**) OSGT, (**g**) BG-OSGT and (**h**) MOSGT when the number of training samples is ten per class.

Table 3. Classification performance using SVM, SuperPCA, CNN, ERS, NG-OSGT, OSGT, BG-OSGT and MOSGT for Houston dataset with ten labeled samples per class as training set.

Class	SVM	SuperPCA	CNN	ERS	NG-OSGT	OSGT	BG-OSGT	MOSGT
C1	93.12	54.12	79.36	77.96	72.94	**93.68**	90.73	90.03
C2	82.31	47.19	**93.44**	50.51	68.25	87.08	87.90	90.29
C3	68.65	98.11	**99.89**	94.76	80.13	89.94	83.75	87.71
C4	82.47	31.90	48.34	45.73	75.60	92.14	**97.56**	94.61
C5	**92.01**	77.09	81.05	78.17	69.06	90.85	90.22	86.38
C6	**94.48**	83.52	62.25	50.00	58.95	82.80	90.95	72.61
C7	67.27	36.41	77.46	78.38	69.14	**78.83**	78.21	76.59
C8	78.97	33.48	52.93	70.87	58.67	82.15	**88.82**	79.47
C9	83.03	37.11	61.18	76.73	70.93	76.44	73.08	**79.56**
C10	63.68	62.50	39.02	63.89	54.62	70.54	**83.60**	81.62
C11	59.16	76.86	54.80	60.65	62.04	70.23	76.75	**80.40**
C12	57.72	50.93	83.78	70.16	60.70	76.78	**88.36**	84.13
C13	40.78	58.00	0.98	59.16	47.62	55.25	70.44	**71.44**
C14	69.95	**100.00**	90.31	90.28	85.97	79.97	97.23	82.57
C15	98.63	81.23	88.38	98.19	76.65	98.04	**98.64**	98.05
OA	75.39	53.52	64.68	68.71	66.37	81.18	**85.39**	83.38
AA	77.38	49.09	64.81	71.34	67.42	81.59	**86.40**	83.36
Kappa	0.73	0.63	0.62	0.66	0.64	0.80	**0.84**	0.82

The second experiment is conducted on the MUUFL Gulfport dataset. Similarly, to further analyze the classification performance of the proposed OSGT method, 10 training samples of each class are randomly selected. The quantitative metrics and classification maps of the compared methods are depicted in Table 4 and Figure 17. When only several training samples are taken for per class, the proposed MOSGT outperforms other comparison methods in terms of visual quality and objective measurement. This demonstrated that the effectiveness of the proposed method in the classification task of HSI and LiDAR data.

Table 4. Classification performance using SVM, SuperPCA, CNN, ERS, NG-OSGT, OSGT, BG-OSGT and MOSGT for MUUFL Gulfpor dataset with ten labeled samples per class as training set.

Class	SVM	SuperPCA	CNN	ERS	NG-OSGT	OSGT	BG-OSGT	MOSGT
C1	96.94	40.34	62.37	96.72	93.38	**96.95**	94.35	94.54
C2	55.16	32.55	**90.21**	50.36	50.03	47.30	52.87	51.27
C3	65.42	27.83	33.79	52.93	66.27	68.71	70.02	**74.71**
C4	55.02	39.94	60.24	56.03	38.13	54.84	55.39	**57.93**
C5	**87.24**	28.55	60.15	69.90	68.47	77.63	73.12	78.49
C6	53.70	**86.14**	4.41	32.13	28.41	40.33	53.72	38.31
C7	39.01	77.22	**78.69**	52.82	41.46	51.29	47.48	56.20
C8	83.14	40.83	59.55	**96.04**	60.70	83.36	88.64	90.41
C9	30.91	36.67	23.48	34.28	12.83	25.64	41.19	**44.60**
C10	13.04	**32.20**	4.34	1.36	1.30	6.85	9.96	8.73
C11	54.31	85.56	58.19	**99.58**	32.46	56.21	86.54	83.57
OA	70.59	38.71	59.22	70.99	59.48	72.67	74.57	**75.63**
AA	57.62	27.45	48.67	58.37	44.86	64.67	61.21	**61.71**
Kappa	0.64	0.48	0.50	0.64	0.5	0.56	0.67	**0.69**

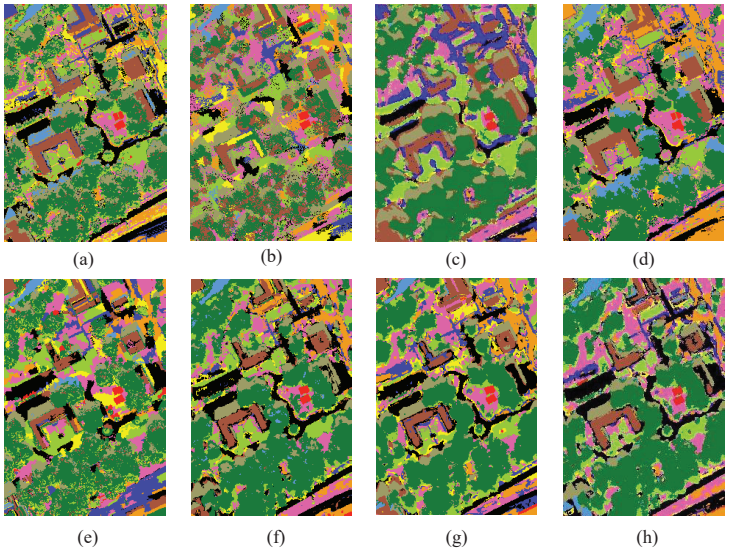


Figure 17. MUUFL Guflport: classification maps obtained by:(**a**) SVM, (**b**) SuperPCA, (**c**) CNN, (**d**) ERS, (**e**) NG-OSGT, (**f**) OSGT, (**g**) BG-OSGT and (**h**) MOSGT when the number of training samples is ten per class.

4.8. Computational Complexity

Table 5 reports the computational time (in seconds) of each component of the proposed OSGT method. The experiments are performed using MATLAB on a computer with a 2.2 GHz CPU and 8 GB of memory. The training size is 10 per class for Houston and MUUFl Gulfport data sets. As presented in Table 5, the main computational cost of the proposed OSGT method is caused by guided filtering operation. The primary reason is that each band of HSI is the operation object of guided filtering. To solve time-consumption problem, we will study how to use graphics processing units (GPUs) to accelerate our algorithm in future developments.

Table 5. Calculation time (in seconds) for different components and guiding images.

Data Set	Different Components			Different Guide Images	
	ERS	**OSGT**	**GF**	**G-PCA**	**G-LiDAR**
Houston	8.69	0.19	33.47	32.94	32.76
MUUFL Gulfport	0.60	0.09	0.94	0.95	0.93

5. Conclusions

In this paper, a OSGT method is proposed for HSI and LiDAR data classification. Specifically, we define homogeneous region fusion between PCs and LiDAR data as a mathematical optimization problem and introduce the gradient transfer model to fuse spectral and DSM information from various superpixel blocks for the first time. Besides, A l^1 total variation minimization is designed to fuse information between the PC and DSM within each superpixel block to accurately describe the observed details. Experimental results on two real datasets indicated that the proposed methods outperforms the considered baseline methods when there are only ten samples per class for training. In the future, injecting the DSM information of LiDAR data into the classification task of HSI by effectively designing a deep convolutional network is a research direction that we focus on.

Author Contributions: B.T., Y.Z. and C.Z. provided algorithmic ideas for this study, designed experiments and wrote the manuscript. S.C. and A.P. participated in the analysis and evaluation of this work. All authors contributed significantly and participated sufficiently to take responsibility for this research study. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61977022, in part by the Science Foundation for Distinguished Young Scholars of Hunan Province under Grant 2020JJ2017, in part by the Key Research and Development Program of Hunan Province under Grant 2019SK2012, in part by the Foundation of Department of Water Resources of Hunan Province under XSKJ2021000-12 and XSKJ2021000-13, in part by the Natural Science Foundation of Hunan Province under Grant 2020JJ4340, and 2021JJ40226, and in part by the Foundation of Education Bureau of Hunan Province under Grant 19A200, 20B257, 20B266, 21B0595 and 21B0590. This work was supported in part by the Open Fund of Education Department of Hunan Province under Grant 20K062. This work was also supported by Junta de Extremadura under Grant GR18060, and by Spanish Ministerio de Ciencia e Innovación through project PID2019-110315RB-I00 (APRISA).

Data Availability Statement: Houston Dataset can be obtained from [9], and MUUFL Gulfport Dataset was obtained from [45,46]. The data generated and analyzed in this study are available on request from the corresponding authors. The data not made public due to follow-up work requirements.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this article.

References

1. Zhou, C.; Tu, B.; Li, N.; He, W.; Plaza, A. Structure-Aware Multikernel Learning for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 9837–9854. [CrossRef]
2. AL-Alimi, D.; Al-qaness, M.; Cai, Z.; Dahou, A.; Shao, Y.; Issaka, S. Meta-Learner Hybrid Models to Classify Hyperspectral Images. *Remote Sens.* **2022**, *14*, 1038. [CrossRef]
3. Xi, J.; Ersoy, O.; Fang, J.; Cong, M.; Wu, T.; Zhao, C.; Li, Z. Wide Sliding Window and Subsampling Network for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 1290. [CrossRef]
4. Wu, L.; Gao, Z.; Liu, Y.; Yu, H. Study of uncertainties of hyperspectral image based on Fourier waveform analysis. In Proceedings of the IGARSS 2004, Anchorage, AK, USA, 20–24 September 2004; Volume 5, pp. 3279–3282.
5. Zhou, C.; Tu B.; Ren, Q.; Chen, S. Spatial Peak-Aware Collaborative Representation for Hyperspectral Imagery Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
6. Gao, R.; Li, M.; Yang, S.; Cho, K. Reflective Noise Filtering of Large-Scale Point Cloud Using Transformer. *Remote Sens.* **2022**, *14*, 577. [CrossRef]
7. Ojogbane, S.; Mansor, S.; Kalantar, B.; Khuzaimah, Z.; Shafri, H.; Ueda, N. Automated Building Detection from Airborne LiDAR and Very High-Resolution Aerial Imagery with Deep Neural Network. *Remote Sens.* **2021**, *13*, 4803. [CrossRef]
8. Gu, Y.; Wang, Q. Discriminative Graph-Based Fusion of HSI and LiDAR Data for Urban Area Classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 906–910. [CrossRef]
9. Debes, C.; Merentitis, A.; Heremans, R.; Hahn, J.; Frangiadakis, N.; Kasteren,T.; Liao, W.; Bellens, R.; Pižurica, A.; Gautama, S. Hyperspectral and LiDAR Data Fusion: Outcome of the 2013 GRSS Data Fusion Contest. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2014**, *7*, 2405–2418. [CrossRef]
10. Saunders, C.; Stitson, M.; Weston, J.; Holloway, R.; Bottou, L.; Scholkopf, B.; Smola, A. Support Vector Machine. *Comput. Sci.* **2002**, *1*, 1–28.
11. Chen, Y. Multiple Kernel Feature Line Embedding for Hyperspectral Image Classification. *Remote Sens.* **2019**, *11*, 2892. [CrossRef]
12. Li, J.; Bioucas-Dias, J.; Plaza, A. Spectral–Spatial Hyperspectral Image Segmentation Using Subspace Multinomial Logistic Regression and Markov Random Fields. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 809–823. [CrossRef]
13. Haut, J.; Paoletti, M. Cloud Implementation of Multinomial Logistic Regression for UAV Hyperspectral Images. *IEEE J. Miniat. Air Space Syst.* **2020**, *1*, 163–171. [CrossRef]
14. Zhong, Y.; Zhang, L. An Adaptive Artificial Immune Network for Supervised Classification of Multi-/Hyperspectral Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 894–909. [CrossRef]
15. Zhang, X.; Shang, S.; Tang, X.; Feng, J.; Jiao, L. Spectral Partitioning Residual Network with Spatial Attention Mechanism for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5507714. [CrossRef]
16. Cui, Y.; Xia, J.; Wang, Z.; Gao, S.; Wang, L. Lightweight Spectral-Spatial Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5510114. [CrossRef]
17. Meng, Z.; Zhao, F.; Liang, M. SS-MLP: A Novel Spectral-Spatial MLP Architecture for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 4060. [CrossRef]
18. Wang, J.; Huang, R.; Guo, S.; Li, L.; Pei, Z.; Liu, B. HyperLiteNet: Extremely Lightweight Non-Deep Parallel Network for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 866. [CrossRef]

19. Fang, L.; Wang, C.; Li, S.; Benediktsson, J. Hyperspectral Image Classification via Multiple-Feature-Based Adaptive Sparse Representation. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 1646–1657. [CrossRef]
20. Ding, Y.; Guo, Y.; Chong, Y.; Pan, S.; Feng, J. Global Consistent Graph Convolutional Network for Hyperspectral Image Classification. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–16. [CrossRef]
21. Zhang, Z.; Liu, D.; Gao, D.; Shi, G. S^3Net: Spectral-Spatial-Semantic Network for Hyperspectral Image Classification with the Multiway Attention Mechanism. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–17.
22. Chen, Y.; Xu, L.; Fang, Y.; Peng, J.; Yang, W.; Wong, A.; Clausi, D. Unsupervised Bayesian Subpixel Mapping of Hyperspectral Imagery Based on Band-Weighted Discrete Spectral Mixture Model and Markov Random Field. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 162–166. [CrossRef]
23. Andrejchenko, V.; Liao, W.; Philips, W.; Scheunders, P. Decision Fusion Framework for Hyperspectral Image Classification Based on Markov and Conditional Random Fields. *Remote Sens.* **2019**, *11*, 624. [CrossRef]
24. Yu, H.; Shang, X.; Song, M.; Hu, J.; Jiao, T.; Guo, Q.; Zhang, B. Union of Class-Dependent Collaborative Representation Based on Maximum Margin Projection for Hyperspectral Imagery Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 553–566. [CrossRef]
25. Su, H.; Yu, Y.; Du, Q.; Du, P. Ensemble Learning for Hyperspectral Image Classification Using Tangent Collaborative Representation. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3778–3790. [CrossRef]
26. Zhong, S.; Chang, C.; Zhang, Y. Iterative Edge Preserving Filtering Approach to Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 90–94. [CrossRef]
27. Wei, Y.; Zhou, Y. Spatial-Aware Network for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 3232. [CrossRef]
28. Khodadadzadeh, M.; Li, J.; Prasad, S.; Plaza, A. Fusion of Hyperspectral and LiDAR Remote Sensing Data Using Multiple Feature Learning. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2015**, *8*, 2971–2983. [CrossRef]
29. Wang, P.; Wang, Y.; Zhang, L.; Ni, K. Subpixel Mapping Based on Multisource Remote Sensing Fusion Data for Land-Cover Classes. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
30. Zhao, X.; Tao, R.; Li, W.; Philips, W.; Liao, W. Fractional Gabor Convolutional Network for Multisource Remote Sensing Data Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–18. [CrossRef]
31. Jahan, F.; Zhou, J.; Awrangjeb, M.; Gao, Y. Inverse Coefficient of Variation Feature and Multilevel Fusion Technique for Hyperspectral and LiDAR Data Classification. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2020**, *13*, 367–381. [CrossRef]
32. Chen, Y.; Li, C.; Ghamisi, P.; Jia, X.; Gu, Y. Deep Fusion of Remote Sensing Data for Accurate Classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1253–1257. [CrossRef]
33. Huang, Q.; Miao, Z.; Zhou, S.; Chang, C.; Li, X. Dense Prediction and Local Fusion of Superpixels: A Framework for Breast Anatomy Segmentation in Ultrasound Image With Scarce Data. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–8. [CrossRef]
34. Jia, S.; Zhan, Z.; Zhang, M.; Xu, M.; Huang, Q.; Zhou, J.; Jia, X. Multiple Feature-Based Superpixel-Level Decision Fusion for Hyperspectral and LiDAR Data Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 1437–1452. [CrossRef]
35. Zhao, W.; Jiao, L.; Ma, W.; Zhao, J.; Zhao, J.; Liu, H.; Cao, X.; Yang, S. Superpixel-Based Multiple Local CNN for Panchromatic and Multispectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4141–4156. [CrossRef]
36. Jiang, J.; Ma, J.; Chen, C.; Wang, Z.; Cai, Z.; Wang, L. SuperPCA: A Superpixelwise PCA Approach for Unsupervised Feature Extraction of Hyperspectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4581–4593. [CrossRef]
37. Zhang, X.; Jiang, X.; Jiang, J.; Zhang, Y.; Liu, X.; Cai, Z. Spectral-Spatial and Superpixelwise PCA for Unsupervised Feature Extraction of Hyperspectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–10. [CrossRef]
38. Liu, M.; Tuzel, O.; Ramalingam, S.; Chellappa, R. Entropy-Rate Clustering: Cluster Analysis via Maximizing a Submodular Function Subject to a Matroid Constraint. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 99–112. [CrossRef]
39. He, K.; Jian, S.; Tang, X. Guided image filtering. In Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–14.
40. Wu, L.; Jong, C. A High-Throughput VLSI Architecture for Real-Time Full-HD Gradient Guided Image Filter. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 1868–1877. [CrossRef]
41. Yang, Y.; Wan, W.; Huang, S.; Yuan, F.; Yang, S.; Que Y. Remote Sensing Image Fusion Based on Adaptive IHS and Multiscale Guided Filter. *IEEE Access* **2016**, *4*, 4573–4582. [CrossRef]
42. Fang, J.; Hu, S.; Ma, X. SAR image de-noising via grouping-based PCA and guided filter. *IEEE J. Syst. Eng. Electron.* **2021**, *32*, 81–91.
43. Draper, N. Applied regression analysis. *Technometrics* **1998**, *9*, 182–183.
44. Chan, T.; Esedoglu, S. Aspects of total variation regularized L1 function approximation. *SIAM J. Appl. Math.* **2005**, *65*, 1817–1837. [CrossRef]
45. Gader, P.; Zare, A.; Close, R.; Aitken, J.; Tuell, G. *Muufl Gulfport Hyperspectral and Lidar Airborne Data Set*; University of Florida: Gainesville, FL, USA, 2013.
46. Du, X.; Zare, A. *Technical Report: Scene Label Ground Truth Map for Muufl Gulfport Data Set*; University of Florida: Gainesville, FL, USA, 2017.
47. Kang, X.; Li, C.; Li, S.; Lin, H. Classification of Hyperspectral Images by Gabor Filtering Based Deep Network. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2018**, *11*, 1166–1178. [CrossRef]

48. Karam, C.; Hirakawa, K. Monte-Carlo Acceleration of Bilateral Filter and Non-Local Means. *IEEE Trans. Image Process.* **2018**, *27*, 1462–1474. [CrossRef] [PubMed]
49. Kang, X.; Li, S.; Benediktsson, J. Feature Extraction of Hyperspectral Images with Image Fusion and Recursive Filtering. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 3742–3752. [CrossRef]
50. Chen, Z.; Jiang, J.; Zhou, C.; Fu, S.; Cai, Z. SuperBF: Superpixel-Based Bilateral Filtering Algorithm and Its Application in Feature Extraction of Hyperspectral Images. *IEEE Access* **2019**, *7*, 147796–147807. [CrossRef]
51. Xu, X.; Li, W.; Ran, Q.; Du, Q.; Gao, L.; Zhang, B. Multisource Remote Sensing Data Classification Based on Convolutional Neural Network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 937–949. [CrossRef]

remote sensing

MDPI

Article

Estimation of Ground PM2.5 Concentrations in Pakistan Using Convolutional Neural Network and Multi-Pollutant Satellite Images

Maqsood Ahmed [1], Zemin Xiao [1] and Yonglin Shen [2,*]

[1] School of Geography and Information Engineering, China University of Geosciences, Wuhan 430074, China; hakromaqsood@cug.edu.cn (M.A.); 1202010855@cug.edu.cn (Z.X.)
[2] National Engineering Research Center of Geographic Information System, China University of Geosciences, Wuhan 430074, China
* Correspondence: shenyl@cug.edu.cn

Abstract: During the last few decades, worsening air quality has been diagnosed in many cities around the world. The accurately prediction of air pollutants, particularly, particulate matter 2.5 (PM2.5) is extremely important for environmental management. A Convolutional Neural Network (CNN) P-CNN model is presented in this paper, which uses seven different pollutant satellite images, such as Aerosol index (AER AI), Methane (CH_4), Carbon monoxide (CO), Formaldehyde (HCHO), Nitrogen dioxide (NO_2), Ozone (O_3) and Sulfur dioxide (SO_2), as auxiliary variables to estimate daily average PM2.5 concentrations. This study estimates daily average of PM2.5 concentrations in various cities of Pakistan (Islamabad, Lahore, Peshawar and Karachi) by using satellite images. The dataset contains a total of 2562 images from May-2019 to April-2020. We compare and analyze AlexNet, VGG16, ResNet50 and P-CNN model on every dataset. The accuracy of machine learning models was checked with Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). The results show that P-CNN is more accurate than other approaches in estimating PM2.5 concentrations from satellite images. This study presents robust model using satellite images, useful for estimating PM2.5 concentrations.

Keywords: deep learning; satellite images; PM2.5; estimation

Citation: Ahmed, M.; Xiao, Z.; Shen, Y. Estimation of Ground PM2.5 Concentrations in Pakistan Using Convolutional Neural Network and Multi-Pollutant Satellite Images. *Remote Sens.* **2022**, *14*, 1735. https://doi.org/10.3390/rs14071735

Academic Editors: Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Received: 18 March 2022
Accepted: 2 April 2022
Published: 4 April 2022

1. Introduction

Particulate matter of a diameter of 2.5 μm (PM2.5) is hazardous for human health, leading to further damage and the destruction of lung function [1–6]. These fine particles are extremely dangerous if they get into the lungs, which might complement the seriousness of COVID-19 infection, and increases the chances of attacks and damage to the respiratory system [7]. Overall, these hazardous pollutants impact human health and produce life-threatening complications in a short period if found in the atmosphere in large concentrations [8]. The research has proven that these particulate matters can potentially affect humans at the genetic level [9].

Various methods have been presented to better explain city-wide air quality, for example, the recent Neighbor legislation and spatial averaging [10,11], to make the most of the limited data gathered by monitoring stations using spatial interpolation. The data sparsity problem is solved by adding monitoring data in most of these systems, which are based on the assumption that air pollution particles diffuse in a spatially continuous manner. However, there are two significant drawbacks of these methods. First, different estimation approaches obtain completely different results. Second, the differences in results are particularly unsatisfactory for raw data with sparse spatial distribution. The air quality detecting network has been optimized by various researchers [12]. For instance, Mei et al. [13] suggest a method to monitor air quality utilizing mobile data. Crowdsourcing computing,

including the use of auxiliary sensors, is rapidly becoming the focus of academic research. Murty et al. [14] suggest a new air pollution monitoring system called CitySense for monitoring air pollutants. In order to obtain data samples using compressed sensing technology, Yu et al. [15] proposed a monitoring strategy that relies on vehicular sensor networks (VSN), which represent a paradigm shift in transportation technology. VSN has the potential to significantly enhance the transportation environment due to the vehicles' infinite power source and the resultant low energy constraints. Li et al. [16] used portable sensors and smartphones to track particulate matter and gas pollutants. However, portable sensors still have limited capability to accomplish the accuracy of monitoring stations accurately. In addition, it takes almost 1 h to obtain the data for PM2.5 measuring equipment; as well, it is also crucial to avoid common issues due to shaking and movement.

Recently, satellite remote sensing has been used in a variety of studies to evaluate air quality [17–24]. For the more accurate methods, an artificial neural network can be utilized as a classifier based on data from road networks and weather data [25]. The deep learning algorithms have achieved significant advancements in image feature learning and have solved numerous challenges in typical computer vision [26]. Image feature-based learning is mainly concerned with the relationships between image characteristics and the index of PM. Liuetal. [27] investigated how air quality relates to image quality. Wang et al. [28] examined air quality by incorporating the association between observed image degradation and PM2.5. Other authors used decision tree in estimating air quality [29]. For example, Zhang et al. [30] used images to calculate air pollution levels with a CNN algorithm. A CNN is a multilayer network structure, whose fundamental structure is comprised of the input layer, convolution layer, pooling layer, fully connected layer and output layer. A convolutional neural network (CNN) is a type of artificial neural network (ANN) that is most typically used to evaluate visual images. It is one of the most widely used types of ANN. This deep learning method can be used to recognize images and videos in a variety of contexts, including recommendation systems, image classification, segmentation, and medical image analysis. The designed CNN was employed to identify photos according to their PM2.5 index via classification. The CNN consists of multiple layers: nine convolutional layers, two pooling layers and two dropout layers, and to overcome the gradient disappearance problem, an enhanced rectified linear unit activation function can be used. Furthermore, the VGG-16CNN model was proposed to evaluate PM2.5 levels [31] on the basis of image-based PM2.5 concentration levels.

According to atmospheric chemistry and physics, the PM2.5 formations are linked to pollutants, such as PM10, CO_2, NO_2 and meteorological variables, also called auxiliary variables, which can be used as input variables for model prediction [32]. Song et al. [33] proposed a statistical model for the estimation of PM2.5 concentration. Their model showed that the concentration of PM2.5 is closely associated with concentrations of NO_2, SO_2, CO and O_3 gaseous pollutants. Therefore, these contaminants can be used as input variables for PM2.5 predictions. Image detection-based air quality research is carried out by combining image processing methods and machine learning approaches, but both have certain weaknesses. For example, the color characteristics of the sky may alter the features utilized in PM2.5 and PM10 concentration detection methods based on visual features from the phone camera image. The sensitivity is excessively high and it is greatly affected by the weather. The detection of PM2.5 and PM10 concentrations based on physical properties may produce pretty good results, but it is only suited for dry air images, which are impacted by meteorological factors. Taking photos from a camera phone have few disadvantages; such as, we can capture photos with high resolution camera in day time; however, in the evening and night time, the quality might be compromised, which does not lead to better results being estimated. Second, it is very inconvenient and difficult to access remotely areas with camera devices; in contrast, satellite images are better to estimate air quality.

This study uses satellite images and employs a novel deep learning-based method for PM2.5 predictions. This technique, such as prediction from satellite images, is not limited by locations and can be suitable to detect air quality at any location. This study uses seven

satellite images (AER AI, CH_4, CO, HCHO, NO_2, O_3 and SO_2) collected by high resolution sensors (TROPOMI) from the sentinel-5p satellite. The method that we used in this study differs from existing methods. It estimates the daily average of PM2.5 concentration using satellite images collected by the TROPOMI sensor of sentinel-5p satellite every day. It can address the weaknesses of present air quality detection technologies and offer fine-grained, low-cost air quality monitoring. The proposed technique can estimate the AQI directly, which is broader and better reflects the air quality. The air quality index (AQI) is a daily indicator that measures the quality of the air at a certain location. It is a way to measure how air pollution affects a person's health during a short period of time (less than 24 h). In short, this study investigates the relationship between PM2.5 concentrations and the concentrations of various pollutants based on satellite images. P-CNN recognizes and extracts patterns and features from input images, and it estimates the daily average of PM2.5 concentrations from these images. This study used four datasets covering Islamabad, Karachi, Lahore and Peshawar city, each dataset contains seven pollutants' images for each day. This paper proposes a deep convolutional neural network model to estimate PM2.5 concentrations from seven given input images. In addition, we also conducted comparative analysis of our proposed model with other three deep learning models on four datasets for more robust results.

This paper is structured in the following way. The second section introduces the study area, datasets and methodology. The third section presents result and discussion of the study, followed by the conclusion and implications in the last section.

2. Materials and Methods

2.1. Study Area and Dataset

The study area we have chosen in this paper is Pakistan. We have taken four metropolitan cities for our experiments such as Karachi, Lahore, Islamabad and Peshawar. Figure 1 shows the study areas and monitoring stations for PM2.5 in Pakistan.

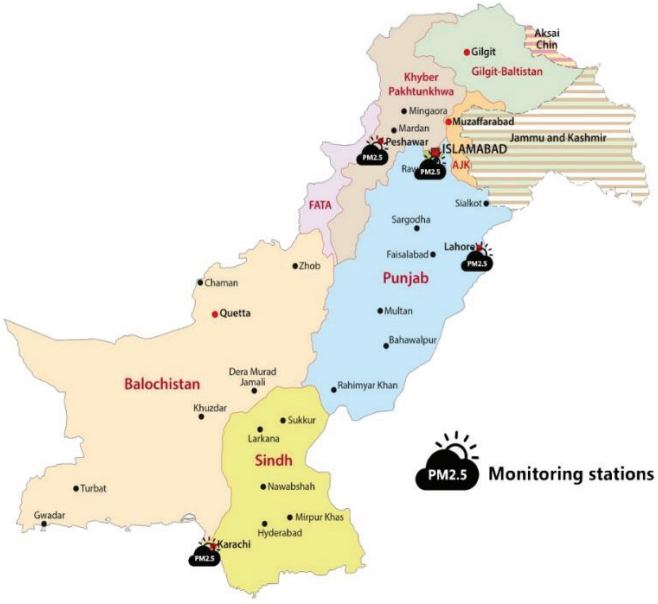


Figure 1. Study area and the distribution of monitoring stations.

There is no openly available library to estimate PM2.5 concentrations from satellite images; therefore, based on sentinel-5p satellite, a multi-input air quality image database was built for each city (Islamabad, Lahore, Peshawar and Karachi). The library contains

2562 images with different PM2.5 levels, which are a collection of scene satellite images at different PM2.5 levels. We used the following steps to create the dataset:

- We collected scene images for each city from the official website [34] from May-2019 to April-2020. Each day contains seven different pollutant images (AER AI, CH_4, CO, HCHO, NO_2, O_3 and SO_2). Table 1 describes the information about the air quality image collection point. One Image cannot cover the concentration of various gases; therefore, each sample is described by taking at least seven satellite images in our research work. The standard single-input CNN architecture is not suitable for our research. Thus, a novel P-CNN model was built to accept seven images as input.

Table 1. Satellite image collection information.

Numbering	Collection Point	Photo Pixels (Px)	Capturing Time Period	Collection Interval
A	Islamabad	3310 × 1573	8:00–9:00 UTC	One per day
B	Peshawar	3310 × 1573	8:00–9:00 UTC	One per day
C	Karachi	3310 × 1573	8:00–9:00 UTC	One per day
D	Lahore	3310 × 1573	8:00–9:00 UTC	One per day

Figure 2 shows the actual satellite images of seven air pollutants with different PM2.5 air quality levels in the image library. Figure 2, such as from A to D shows different days, while I, II, III, IV, V, VI, and VII are seven different pollutant images by sentinel-5p satellite for same day. I represents concentration of AER AI pollutant in single day, while II illustrates CH_4 pollutant concentration for same day. III number image is about CO concentration. IV image is about HCHO pollutant concentration. V, VI and VII images are examples of NO, O_3, and SO_2, respectively.

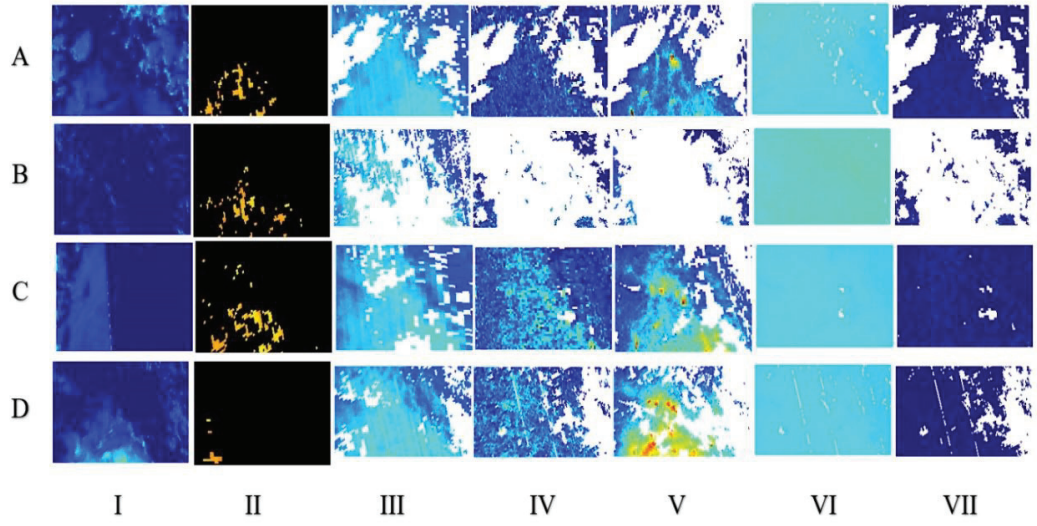


Figure 2. Example of 4 days of seven different satellite input images in dataset.

Real-time monitoring stations across main cities of Pakistan, such as Islamabad, Lahore, Karachi and Peshawar, measure air quality levels then upload them on the website for the open access. Figure 1 shows the location of the monitoring stations. PM2.5 hourly real-time data were obtained from the official website [35]. Since PM2.5 concentration data are measured hourly by the monitoring stations for each city, we converted the 24-hour data into a daily average to train our model. For the model training, 70% of the images were randomly selected for training and 30% for testing purposes. Furthermore, to prevent

the model from overfitting and improve model accuracy and robustness, we strengthened the dataset training process with the minimal number of samples in the training dataset in the following ways.

(1) Randomly Image Rotation between [0, 360] degrees.
(2) Scale the image at random between [0.8, 1] coefficients.
(3) Size of each auxiliary input pollutant image is adjusted to 300 × 300, and then normalized to [0, 1].

2.2. Convolutional Neural Network (CNN)

CNN, firstly proposed by LeCun et al. [36] for recognition of handwritten digits, has been widely successful in the areas of image detection, segmentation, and identification tasks [37–42]. CNN has shown its remarkable capacity to classify large-scale images. It consists of three-layers: convolutional layers, pooling layers and fully connected layers. The essential layers in CNN are the convolutional and pooling layers. The convolution layers are used to extract features with numerous filters by convolving image regions. As the layers expand, the CNN gradually understands the image. The pooling layers lower the dimensions of output maps from the convolutional layers and avoid overfitting. The number of neurons, parameters and connections in the CNN model is substantially less through these two levels. Thus, CNNs are much more effective than Backpropagation (BP) neural networks with correspondingly sized layers.

2.3. Architecture of P-CNN

Based on the standard CNN architecture, we have proposed a model named P-CNN. The model is employed to estimate PM2.5 concentrations and acquire a preferable result on the dataset. Figure 3 shows the entire model of CNN architecture.

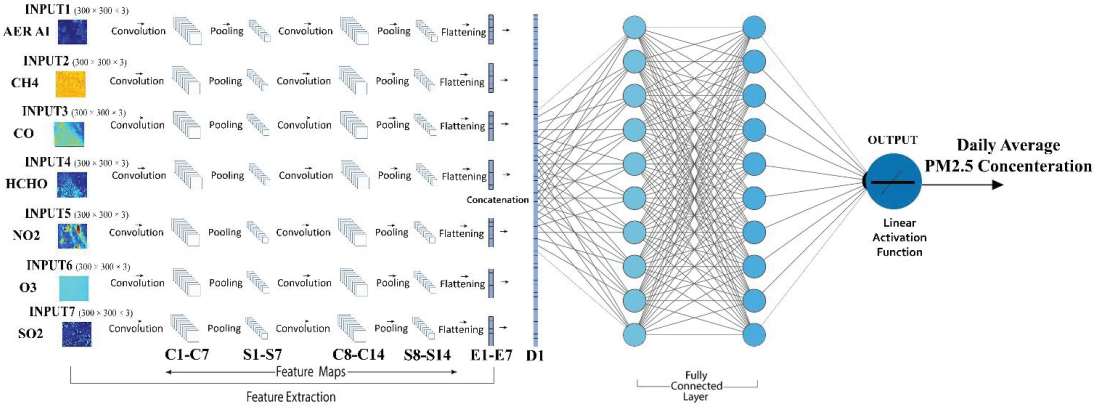


Figure 3. PM2.5 Concentration Estimation Model.

The convolutional layers C1–C7 filter seven 300 × 300 × 3 input images with 32 kernels of size 4 × 4 × 3 with the stride of 1 pixel. The stride of pooling layers S1–S7 is 2 pixels. C8–C14 filter with 16 kernels of size 4 × 4 × 3 with the stride of 1 pixel. The stride of pooling layers S8–S14 is 2 pixels, and the dropout is applied to the output of S8–S14, which has been flattened (E1–E4). D1 is the concatenation of the previous flattened E1–E4. The fully connected layer FC1 has ten neurons, FC2 has ten neurons, and FC3 has one. The activation of the output layer is a linear function.

A high-level neural networks API called "Keras" is used to implement the model [16]. All of the experiments were carried out on an Ubuntu Kylin 14.04 server equipped with a 3.40 GHz i7-3770 CPU (16 GB RAM) and a GTX 1070 graphics card (8 GB memory). The original image has a resolution of 3310 × 1575 pixels, which needs be lowered in order to

fit into the GPU memory. All of the original images are scaled to 300 × 300 pixels, and then the value of per-pixel is divided by 255. In addition, images should be normalized and standardized before being fed into model in order to achieve rapid convergence. A randomization process is used to ensure that the model is not influenced by the sequence in which photographs are input. Both the sequence of samples and the seven images corresponding to each sample should be randomized. The convolutional neural network training procedure is divided into two steps. The first is called forward propagation, and the second is called backward propagation.

2.4. Forward Propagation

Data are transmitted from the input layer to the output layer by a sequence of operations that include convolution, pooling and fully connected. Each convolutional layer employs trainable kernels in order to filter the results of the preceding layer followed by activation function to build the output feature map.

In a general way, the procedure is as follows:

$$x_j^e = f\left(\sum_{i \in M_j} x_i^{e-1} * k_{ij}^e + b_j^e\right) \tag{1}$$

where M_j denotes the collection of input maps we choose. b is the bias that is applied to all output map. k indicates the kernels, the weight of the row "i" and column "j" in each kernel is represented by the k_{ij}^e. Using a kernel map, the outputs of surrounding neurons are summarized by the pooling layer, which is the operation of the pooling layer.

$$x_j^e = f\left(\beta_j^e \, down\left(x_i^{e-1}\right) + b_j^e\right) \tag{2}$$

where β denotes multiplicative bias and b indicates additive bias, "$down$" is a subsampling function that uses the max-pooling algorithm [43]. The reason why we chose max-pooling over mean pooling is that the latter makes it impossible to identify critical information such as the edges of objects, whereas the former selects the most active neuron of each region in feature maps, which is more efficient [44]. As a result, it is easier to extract useful features when using max-pooling. In a multilayer perceptron, the fully connected layer is equivalent to the hidden layer. The activation function "linear" for output layer was employed for regression [45], which is given below by

$$f(x) = ax \tag{3}$$

Any constant value can be for variable "a". A derivative of $f(x)$ in this case is not zero, but is equal to the constant employed. Notably, the gradient does not equal zero, but rather a constant number that is independent of the input value x, which indicates that the weights and biases will be updated throughout the backpropagation phase, despite the fact that the updating factor will remain the same.

2.5. Backward Propagation

Backward propagation adjusts parameters by using stochastic gradient descent (SGD) in order to reduce the disparity between the anticipated outcome and the actual outcome. For the purpose of avoiding overfitting, L_1 and L_2 regularization is used.

$$C = C_0 + \frac{\lambda}{n} \sum_w |w| \tag{4}$$

where C_0 represents loss in the formula (4). The formula for L_2 is given by below

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2 \tag{5}$$

This paper uses a weight of 0.0001 for L1 and L2 regularization. Dropout is also used to prevent overfitting [46], and its value is set to 0.1. The SGD algorithm calculates the gradients and modifies the coefficients or weights. It can be stated in the following way:

$$\delta_x = w_{x+1} + \left(\sigma'\left(w_{x+1} \cdot c_x + b_{x+1}\right) \circ up\left(\delta_x + 1\right)\right) \tag{6}$$

$$\Delta w_x = -n \cdot \sum_{ij}(\delta_x \circ down(S_{x-1})) \tag{7}$$

where δ_x denotes the sensitivities of each unit to fluctuations of the bias b, and $\circ$ represents the element-wise multiplication. An upsampling procedure is represented by the $up()$, and subsampling operation is represented by the $down()$. The updated weight is denoted by w, and n represents the learning rate.

2.6. Evaluation Metrics

The following evaluation measures, Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), were employed in this work to complete the quantitative assessment of the constructed P-CNN model's capabilities.

MAE is a model assessment statistic that is commonly employed in regression models. It is a metric for estimating the average discrepancy between estimates and actual results. It is used to estimate the machine learning model's accuracy.

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{n}-0}^{\mathrm{N}} |\, \mathrm{o_n} - \mathrm{p_n} \,| \tag{8}$$

The Root Mean Square Error (RMSE) is a commonly used metric for determining how well a model predicts quantitative data. Here, RMSE calculates the error between actual (station value) and predicted value (model's predicted value).

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{\mathrm{i}=0}^{\mathrm{n}-1}(\mathrm{y_i} - \mathrm{f_i})^2}{\mathrm{n}}} \tag{9}$$

MAPE means absolute percentage error and is a statistical indicator used for prediction. The "accuracy" of this measurement is expressed as a percentage. It is possible to determine for each period the average absolute percent error, which is deducted from the actual numbers, and then the outcome is divided by actual values. However, the larger the concentration, the bigger the absolute inaccuracy in the forecast. As a result, we anticipate that the MAPE will be able to offer the most accurate forecasts among models.

$$\mathrm{MAPE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=0}^{\mathrm{n}-1} \left| \frac{\mathrm{y_i} - \mathrm{f_i}}{\mathrm{y_i}} \right| \times 100 \tag{10}$$

3. Results

AlexNet, VGG16, ResNet50 and P-CNN were all evaluated for their prediction abilities using three different indicators. They are Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). Table 2 displays MAE results for Lahore, Karachi, Peshawar and Islamabad after applying different machine learning models. When we applied AlexNet on the datasets, a 34.464 average value was achieved, which was reduced 5.113 using ResNet50. VGG16 also decreased the 7.723 MAE value after ResNet50. After applying the P-CNN model on the datasets, 6.475 MAE reduced, and its average value for each city was calculated as 15.152, which is a really good result. Table 3 shows the RMSE values for different cities with different models. AlexNet achieved a 49.445 RMSE average value for all cities, and 12.082 was reduced after applying ResNet50. VGG16 also helped to reduce the 9.079 RMSE value, and 8.726 RMSE decreased after applying P-CNN, and its average value was 19.557. Table 4 reveals results for MAPE. For the

average value for all cities after using the AlexNet model, we achieved 43.932. After employing ResNet50, the 7.373 MAPE value decreased. VGG16 also decreased the 11.990 MAPE value. Lastly, P-CNN reduced 9.403 MAPE after VGG16, and its average value for all cities was 15.167. All of these metrics show that P-CNN is superior to other models.

Table 2. MAE results for all cities using AlexNet, VGG16, ResNet50 and P-CNN.

City	AlexNet	ResNet50	VGG16	P-CNN
Karachi	32.343	28.187	19.554	17.123
Lahore	29.843	30.214	21.240	14.205
Peshawar	37.449	27.345	22.145	18.280
Islamabad	38.221	31.657	23.572	11.003
Average	34.464	29.350	21.627	15.152

Table 3. RMSE results for all cities using AlexNet, VGG16, ResNet50 and P-CNN.

City	AlexNet	ResNet50	VGG16	P-CNN
Karachi	56.322	37.299	29.368	22.084
Lahore	47.917	39.239	24.431	20.835
Peshawar	50.329	32.302	31.502	18.743
Islamabad	43.215	40.611	27.834	16.566
Average	49.445	37.362	28.283	19.557

Table 4. MAPE results for all cities using AlexNet, VGG16, ResNet50 and P-CNN.

City	AlexNet	ResNet50	VGG16	P-CNN
Karachi	45.954	40.223	22.838	14.419
Lahore	42.390	37.901	25.949	12.394
Peshawar	47.987	35.025	21.494	17.200
Islamabad	39.399	33.092	28.001	16.657
Average	43.932	36.560	24.570	15.167

Figure 4 depicts a comparison of the actual values and projected values in a time series graph obtained by applying AlexNet (a), VGG16 (b), ResNet50 (c) and our proposed model P-CNN (d) to a testing dataset for Karachi city. In this figure, the P-CNN obtained values that were more closely aligned with the observed values than AlexNet, VGG16 and ResNet50. According to performance indicators, our P-CNN performs much better than other models in terms of predicting of PM2.5 concentrations. When AlexNet was used to the Karachi testing dataset, it produced the following results: MAE (32.343), RMSE (56.322) and MAPE (45.954). The ResNet50 model obtained the following metrics: MAE (28.187), RMSE (37.299) and MAPE (40.223). VGG16 achieved MAE (19.554), RMSE (29.368) and MAPE (22.838). In the same testing dataset for Karachi, we implemented our proposed model P-CNN and obtained the best results, such as MAE (17.123), RMSE (22.084) and MAPE (14.149). Figure 5 shows the difference between the actual and predicted values after applying the same models to Lahore city. The graphs clearly demonstrate that P-CNN (d) outperformed the other models. AlexNet (a) determined the MAE, RMSE and MAPE for Lahore city (29.843, 47.917 and 42.390). ResNet50 (c) achieved (30.214, 39.239 and 37.901). VGG16 attained (b) (21.240, 24.431 and 24.431). However, while assessing the performance of models for predicting PM2.5 concentration, P-CNN (d) achieved the lowest MAE, RMSE and MAPE (14.205, 20.835 and 12.394). The actual and estimated outcomes for Peshawar city are depicted in Figure 6. The graph clearly demonstrates that the P-CNN (d) estimated values more accurate than AlexNet (a), ResNet50 (c) and VGG16 (b). In addition, performance metrics revealed too that P-CNN (d) outperformed all other models. AlexNet computed MAE, RMSE and MAPE (37.449, 50.329 and 47.987), ResNet50 (27.345, 32.302 and 35.025), VGG16 (22.145, 31.502 and 21.494) and P-CNN (18.280, 18.743

and 17.200). Figure 7 provides a time series graph of the observed and predicted values for Islamabad city. It shows that P-CNN (d) is more accurate in predicting PM2.5 concentrations when compared with the other deep learning models (a), (b) and (c). Testing dataset for Islamabad contains three months of daily average of PM2.5 concentration. After applying performance indicators on Islamabad city, MAE, RMSE and MAPE achieved 38.221, 43.215 and 39.399 by AlexNet; 31.657, 40.611 and 33.092 by ResNet50; 23.572, 27.834 and 28.001 by VGG16; and 11.003, 16.566 and 16.657 by P-CNN. All of these figures and performance metrics clearly demonstrate that P-CNN outperforms other deep learning models, such as AlexNet, ResNet50 and VGG16, in terms of predicting PM2.5 concentrations accurately in Karachi, Lahore, Peshawar and Islamabad.

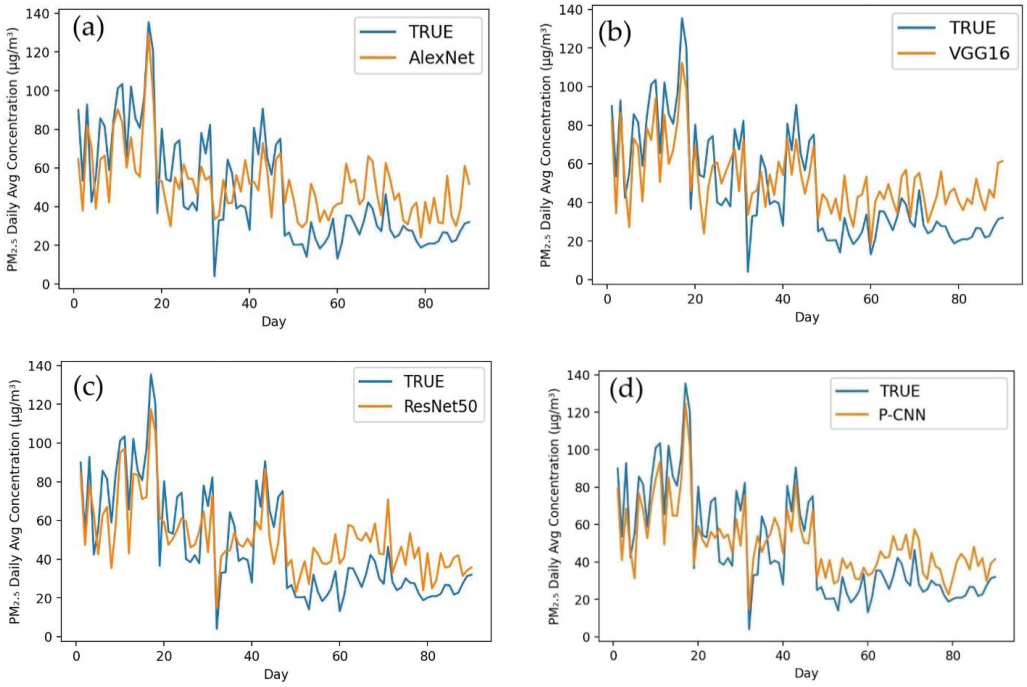


Figure 4. Time series of observed values and predicted values of models for Karachi. (**a**) AlexNet, (**b**) VGG16, (**c**) ResNet50, (**d**) P-CNN.

Consequently, to conduct further testing efficiency of our developed model, P-CNN, we trained a model on one city dataset, and tested on all remaining cities. After training the model on Islamabad, as seen in Figure 8, it can be used to predict PM2.5 concentrations in a number of different cities, such as Karachi, Lahore and Peshawar. Figure 8 clearly demonstrates that the P-CNN predicted values for Karachi, Lahore and Peshawar are extremely close to the real values. The proposed model for predicting PM2.5 concentrations was also trained on a dataset from Karachi and evaluated on datasets from other cities such as Lahore, Peshawar and Islamabad (as shown in Figure 9). The results indicated that the model, which was trained on the Karachi dataset, can be applied to Lahore, Peshawar and Islamabad. It was also found that training a model with Lahore data, can accurately predict PM2.5 concentrations for other cities such as Islamabad, Karachi and Peshawar (see Figure 10). According to Figure 11, using Peshawar as a training dataset, our model is able to predict the concentrations of PM2.5 in other cities such as Islamabad, Lahore and Karachi. These results proved that our proposed P-CNN model also can be applied to other cities after being trained on a single city. Overall, these results demonstrate that P-CNN model is useful in predicting PM2.5 concentrations with satellite images.

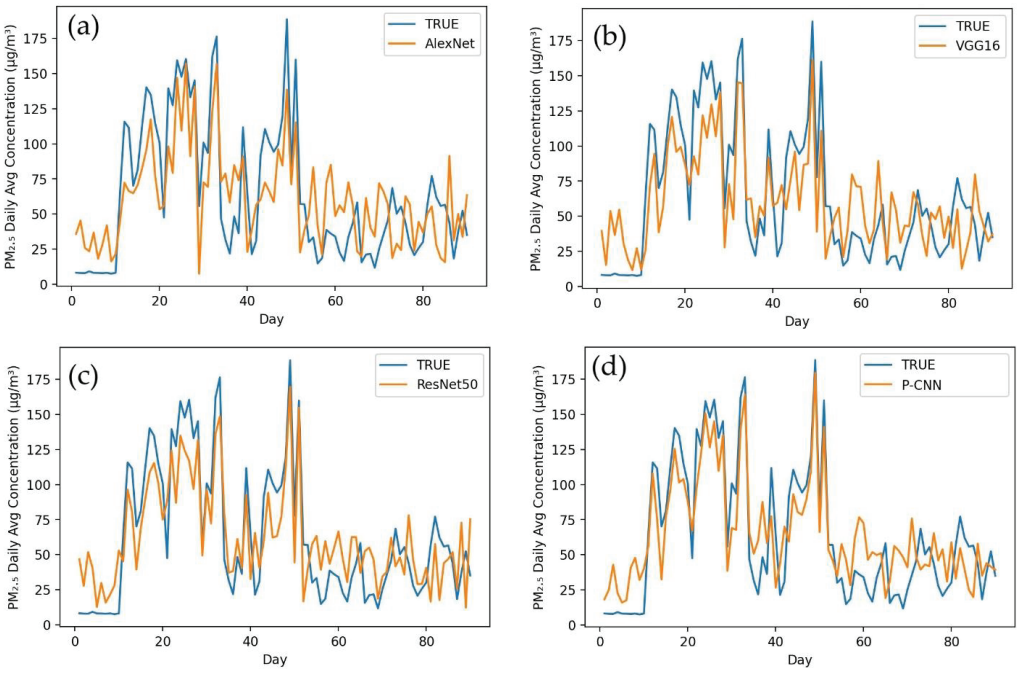


Figure 5. Time series of observed values and predicted values of models for Lahore. (**a**) AlexNet, (**b**) VGG16, (**c**) ResNet50, (**d**) P-CNN.

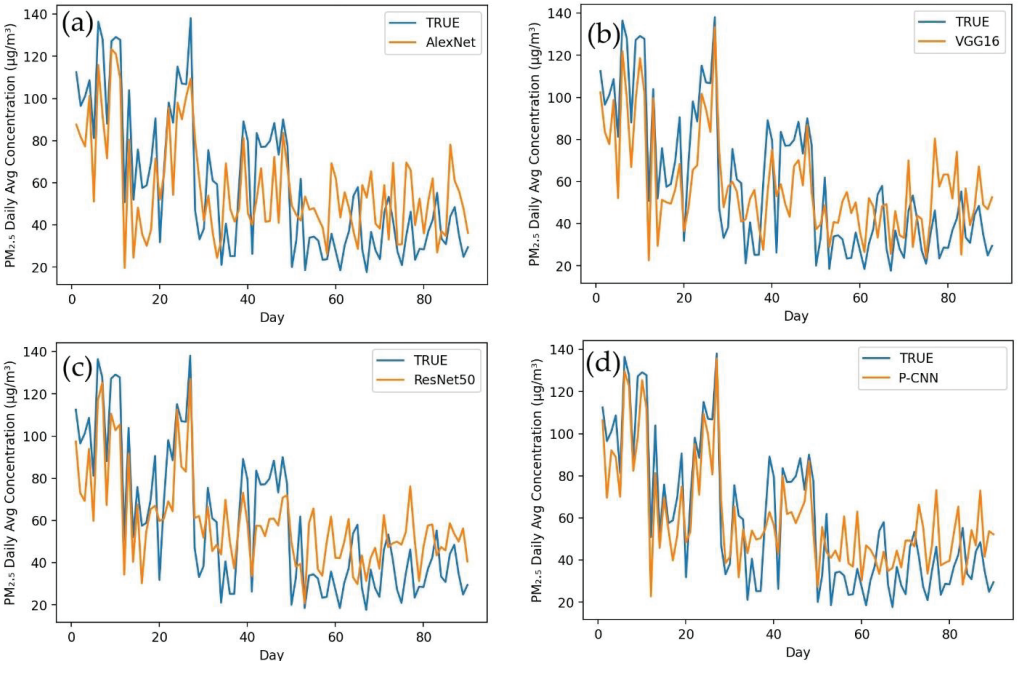


Figure 6. Time series of observed values and predicted values of models for Peshawar. (**a**) AlexNet, (**b**) VGG16, (**c**) ResNet50, (**d**) P-CNN.

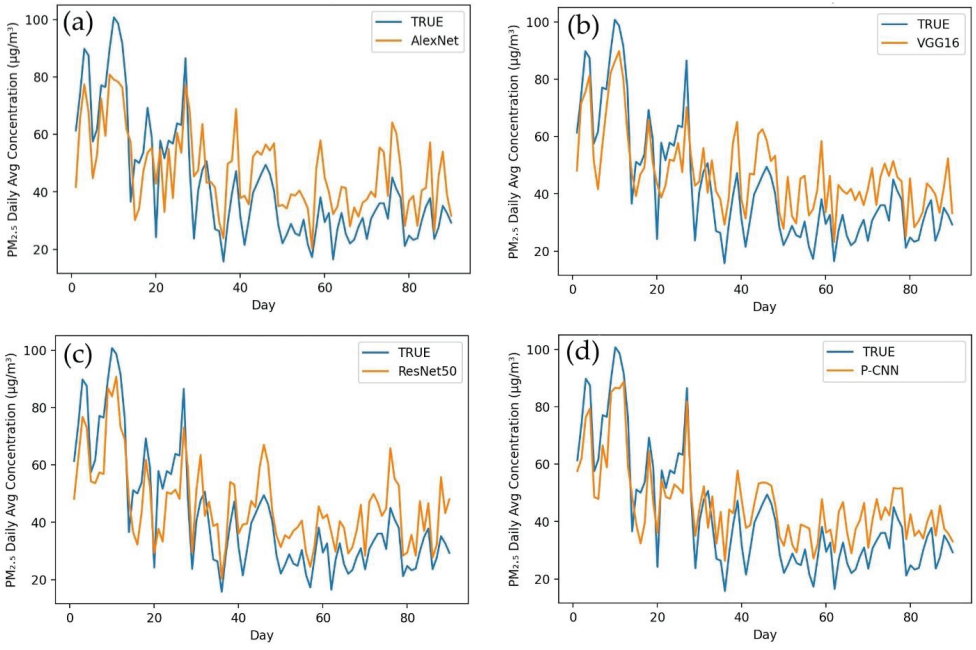


Figure 7. Time series of observed values and predicted values of models for Islamabad. (**a**) AlexNet, (**b**) VGG16, (**c**) ResNet50, (**d**) P-CNN.

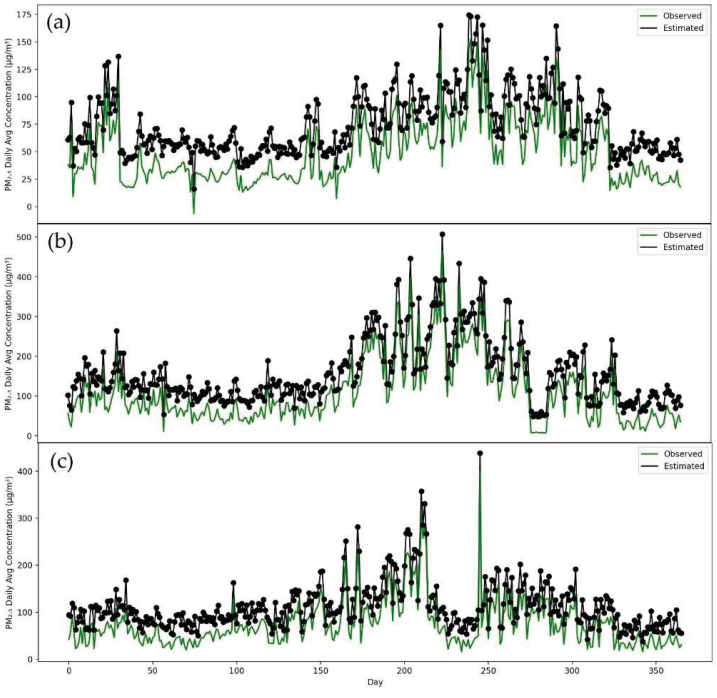


Figure 8. Time series of station values and predicted values by P-CNN model (**a**) Karachi, (**b**) Lahore, (**c**) Peshawar.

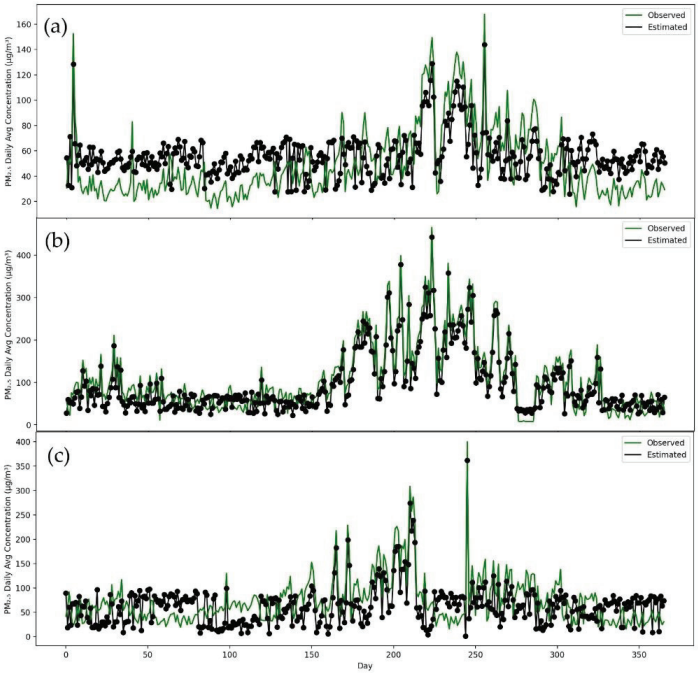


Figure 9. Time series of station values and predicted values by P-CNN model. (**a**) Islamabad, (**b**) Lahore, (**c**) Peshawar.

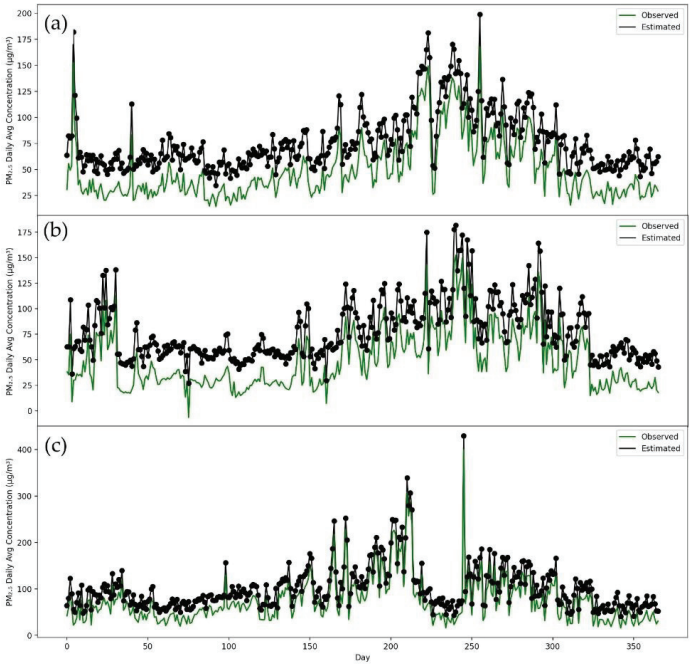


Figure 10. Time series of station values and predicted values by P-CNN model. (**a**) Islamabad, (**b**) Karachi, (**c**) Peshawar.

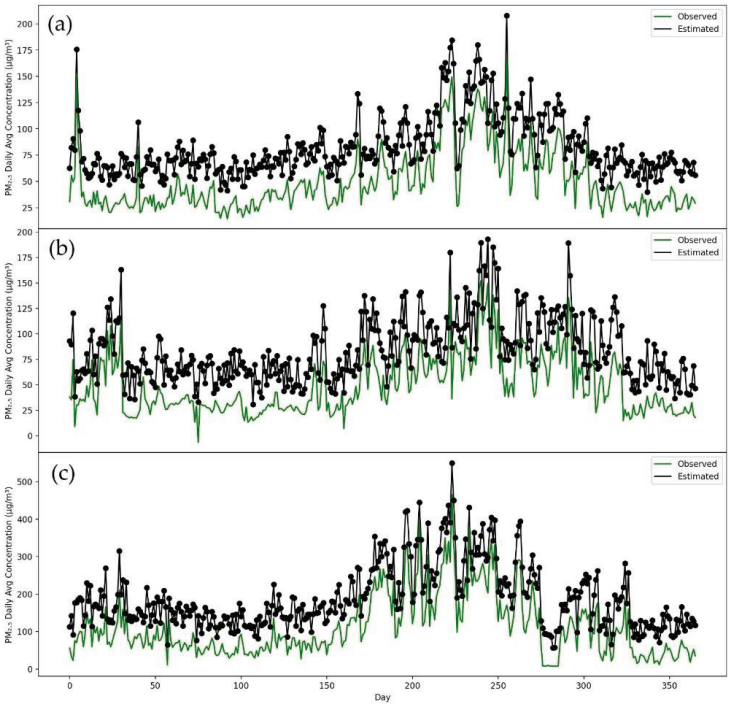


Figure 11. Time series of station values and predicted values by P-CNN model. (**a**) Islamabad, (**b**) Karachi, (**c**) Lahore.

4. Discussion

This study adopts seven inputs to estimate PM2.5 concentrations in four cities, namely, Pakistan, Islamabad, Lahore, Karachi and Peshawar. The findings revealed that seven input pollutants (AER AI, CH_4, CO, HCHO, NO_2, O_3 and SO_2) are closely linked with PM2.5. The existing studies have used different approaches for PM2.5 estimation. Li et al. [47] uses transmission and depth matrices to estimate haze levels. As a proxy for PM2.5, two datasets were utilized for the evaluation. The authors used 8761 photographs in the PM2.5 datasets, and the stated Absolute Spearman correlation is 40.83%. PM2.5's dataset contains three classes: HeavyHaze, LightHaze and NonHaze and the stated correlation is 89.05%. Zhang et al. [48] proposed deep learning method to classify the camera images according to AQI-levels; there were six classes: good, moderate, Unhealthy for Sensitive Groups, Unhealthy, Very Unhealthy and Hazardous. The applied method was tested on the dataset and achieved 74.0% accuracy. Both these studies have developed deep learning models for classification purpose; however, we proposed a novel P-CNN approach, which uses seven auxiliary input satellite images and estimates actual real number, PM2.5 concentrations. Estimating PM2.5 concentrations differs from classifying, segmenting or recognizing objects based on attributes such as color or texture. We tested P-CNN model on four different datasets using statistics metrics. We achieved satisfactory values of MAE (15.152), RMSE (19.557) and MAPE (15.167) using P-CNN model. Furthermore, in estimating PM2.5 concentrations, the results showed that the P-CNN method provides better results. For instance, the advantage of using this model helps to cover remote areas for estimating air quality.

There are various reasons that compared to Islamabad and Peshawar, the air quality in Lahore and Karachi is far worse. Peshawar and Islamabad are smaller and less populated than Lahore and Karachi city. Islamabad and Peshawar city have less public transit than Lahore and Karachi. The number of industries and construction sites are also less in Islam-

abad and Peshawar. Lahore and Karachi have a greater ratio of growing urbanization than Peshawar and Islamabad. On the other hand, Lahore is one of second-largest metropolitan city of Pakistan, with a population of 11 million residents, and has topped the daily rankings of the world's most polluted cities for the second time this year. Tree cover in Lahore has declined significantly over the previous 15 years as a result of an ambitious effort to develop highways, bridges and tunnels. Increasing population, industry, deplorable conditions of municipal utilities, and traffic congestion are the primary sources of air pollution in Karachi city. Furthermore, environmental issues have increased as a result of rapid urbanization such as sewage system inadequacies, overcrowding, inadequate transportation and uncontrolled growth, particularly in Karachi. Air pollution is also exacerbated by industrial pollutants, waste burning, house fires, and other particulates. However, it appears that neither the government nor environmental organizations are taking this matter seriously or responding quickly enough. Similarly, an increase in population accelerates agriculture and industrial production, resulting an increase in waste [49]. Government can help relevant industries by providing green credit funds for the eco-friendly environment, which helps the business community to accelerate green technology and research and development. Pakistan, being a developing economy, suffers huge losses due to environmental problems. During the period between 1999 and 2018, the country spent around USD 3.8 billion to fight against environmental issues in Karachi, Lahore and Peshawar [50]. The water- and land-based ecosystems are being demolished, and unplanned urban structure have damages environment badly. This implies that poor socioeconomic systems cause environmental degradation. Lahore city is the second metropolitan city in Pakistan, covering 2233 manufacturing firms [51]. Lahore is regarded as one of the most developed cities in socioeconomic perspectives. However, some factors, such as industrial waste, poor sanitation systems and lack of urban planning, are barriers to environmental quality. Compared to Karachi and Lahore city, Islamabad is a well-planned city, with the transportation and construction sectors having been developed. On the other hand, Peshawar city is also one of the important hubs in Pakistan. Urban sprawl, deforestation and the burning of contaminated fuel have proved to be the drivers of greenhouse gas emissions [52,53].

Overall, the poor socioeconomic status of these cities has prevented efforts to maintain the ecosystem. Poor infrastructure, dense population and dependency on traditional cook stoves can increase the CO_2, PM2.5 and other greenhouse gas emissions. The findings of Mehmood et al. [53] revealed that most of the households in rural areas of Pakistan burn wood, straw, animal dung and crops for cooking purpose, indicating that the most of the households are dependent on contaminated fuels. Moreover, cooking practices with contaminated fuel have the direct association with PM2.5 concentrations [54]; thus, the government should promote clean energy, provide modern cook stoves and reduce fossil fuel consumption to mitigate PM2.5 and other greenhouse gas emissions in Pakistan.

All four cities (Lahore, Peshawar, Islamabad and Karachi) from 1 January to 31 December 2017, had PM2.5 concentrations above than the standard recommendation (10 mg/m^3). According to AQI rankings of the world's most polluted cities, Lahore was ranked at number six, while Karachi was ranked at number sixteen, with AQI levels of 170 and 155, respectively [55]. Most recently, Lahore ranked as world's most polluted city [56]. Hence, we need immediately the finest and most effective tools and methods to analyze, understand and estimate air quality properly. Our proposed deep learning model for estimating PM2.5 concentrations is efficient and cost saving. We do not need to deploy physical measurement tools in each city to calculate air quality. Using portable devices (laptops, mobiles, etc.), PM2.5 concentrations for any city can be estimated using our deep learning model. Pudasaini et al. [57] had proposed a model to estimate PM2.5 concentration from photographs. However, in order to estimate PM2.5 concentrations, we would need to travel to the site area and snap a picture of it using a mobile phone. However, in our method, we need only chose a city to predict PM2.5 concentrations on portable device anywhere. Thus, this study suggests a reliable and effective way of estimating PM2.5 concentrations.

5. Conclusions

This paper proposes a deep learning P-CNN model for PM2.5 concentrations. This model mainly uses deep convolutional neural networks to extract feature representation information related to PM2.5 in satellite images to estimate PM2.5 concentration levels. We also performed comparative analysis of our constructed model with other deep learning models such as AlexNet, VGG16 and ResNet50 on four different datasets (Karachi, Lahore, Peshawar and Islamabad). The study performed different measures to analyze the model's accuracy. In this regard, MAE, RMSE and MAPE were used as accuracy metrics. The experimental results demonstrated that the P-CNN model is more suitable for predicting PM2.5 concentrations than other models. The results confirmed that the PM2.5 concentrations our model predicts from satellite images are closely related with actual results. Any future research should focus on finding ways to make the model more accurate, as well as to focus on seasonal-wise $PM_{2.5}$ estimations. Although, the model provides better results, some limitations cannot be avoided. Based on available datasets, we used the samples between May-2019 to April-2020. This study focuses on four cities of Pakistan; future study should find large datasets and use more cities, which will give better results.

Author Contributions: Conceptualization, M.A.; methodology, M.A.; software, M.A.; validation, M.A.; formal analysis, Y.S.; investigation, Y.S.; writing—original draft preparation, M.A.; writing—review and editing, Z.X. and Y.S.; visualization; supervision, Z.X. and Y.S.; funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is jointly supported by grants from the National Key Research and Development Program of China (Grant No.: 2020YFB2103403), and the State Key Laboratory of Resources and Environmental Information System.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Xing, Y.-F.; Xu, Y.-H.; Shi, M.-H.; Lian, Y.-X. The impact of PM2.5 on the human respiratory system. *J. Thorac. Dis.* **2016**, *8*, E69. [PubMed]
2. Lewis, T.C.; Robins, T.G.; Dvonch, J.T.; Keeler, G.J.; Yip, F.Y.; Mentz, G.B.; Lin, X.; Parker, E.A.; Israel, B.A.; Gonzalez, L. Air pollution–associated changes in lung function among asthmatic children in Detroit. *Environ. Health Perspect.* **2005**, *113*, 1068–1075. [CrossRef] [PubMed]
3. Bos, I.; Jacobs, L.; Nawrot, T.S.; De Geus, B.; Torfs, R.; Panis, L.I.; Degraeuwe, B.; Meeusen, R. No exercise-induced increase in serum BDNF after cycling near a major traffic road. *Neurosci. Lett.* **2011**, *500*, 129–132. [CrossRef] [PubMed]
4. Jacobs, L.; Nawrot, T.S.; De Geus, B.; Meeusen, R.; Degraeuwe, B.; Bernard, A.; Sughis, M.; Nemery, B.; Panis, L.I. Subclinical responses in healthy cyclists briefly exposed to traffic-related air pollution: An intervention study. *Environ. Health* **2010**, *9*, 64. [CrossRef]
5. Bhatnagar, A. Environmental cardiology: Studying mechanistic links between pollution and heart disease. *Circ. Res.* **2006**, *99*, 692–705. [CrossRef]
6. Valavanidis, A.; Fiotakis, K.; Vlachogianni, T. Airborne particulate matter and human health: Toxicological assessment and importance of size and composition of particles for oxidative damage and carcinogenic mechanisms. *J. Environ. Sci. Health Part C* **2008**, *26*, 339–362. [CrossRef]
7. Kumar, S.; Mishra, S.; Singh, S.K. Deep Transfer Learning-based COVID-19 prediction using Chest X-rays. *J. Health Manag.* **2021**, *23*, 730–746. [CrossRef]
8. Schwartz, J.; Dockery, D.W.; Neas, L.M. Is daily mortality associated specifically with fine particles? *J. Air Waste Manage. Assoc.* **1996**, *46*, 927–939. [CrossRef]
9. Graff, D.W.; Schmitt, M.T.; Dailey, L.A.; Duvall, R.M.; Karoly, E.D.; Devlin, R.B. Assessing the role of particulate matter size and composition on gene expression in pulmonary cells. *Inhal. Toxicol.* **2007**, *19*, 23–28. [CrossRef]
10. Schwartz, J. Lung function and chronic exposure to air pollution: A cross-sectional analysis of NHANES II. *Environ. Res.* **1989**, *50*, 309–321. [CrossRef]
11. Chestnut, L.G.; Schwartz, J.; Savitz, D.A.; Burchfiel, C.M. Pulmonary function and ambient particulate matter: Epidemiological evidence from NHANES I. *Arch. Environ. Health Int. J.* **1991**, *46*, 135–144. [CrossRef] [PubMed]

12. Li, L.; Zhai, C.Z.; YU, J.Y. A Review of Domestic and Overseas Research on Air Quality Monitoring Networks Designing. *Environ. Monit. China* **2012**, *4*, 1–4.
13. Mei, S.; Li, H.; Fan, J.; Zhu, X.; Dyer, C.R. Inferring air pollution by sniffing social media. In Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), Beijing, China, 17–20 August 2014; pp. 534–539.
14. Murty, R.N.; Mainland, G.; Rose, I.; Chowdhury, A.R.; Gosain, A.; Bers, J.; Welsh, M. Citysense: An urban-scale wireless sensor network and testbed. In Proceedings of the 2008 IEEE Conference on Technologies for Homeland Security, Waltham, MA, USA, 12–13 May 2008; pp. 583–588.
15. Yu, X.; Liu, Y.; Zhu, Y.; Feng, W.; Zhang, L.; Rashvand, H.F.; Li, V.O.K. Efficient sampling and compressive sensing for urban monitoring vehicular sensor networks. *IET Wirel. Sens. Syst.* **2012**, *2*, 214–221. [CrossRef]
16. Li, L.; Zheng, Y.; Zhang, L. Demonstration abstract: PiMi air box—A cost-effective sensor for participatory indoor quality monitoring. In Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, Berlin, Germany, 15–17 April 2014; pp. 327–328.
17. Gupta, P.; Christopher, S.A.; Wang, J.; Gehrig, R.; Lee, Y.C.; Kumar, N. Satellite remote sensing of particulate matter and air quality assessment over global cities. *Atmos. Environ.* **2006**, *40*, 5880–5892. [CrossRef]
18. Padayachi, Y.R. Satellite Remote Sensing of Particulate Matter and Air Quality Assessment in the Western Cape, South Africa. 2016. Available online: https://ukzn-dspace.ukzn.ac.za (accessed on 27 February 2022).
19. Chung, Y.S. Air pollution detection by satellites: The transport and deposition of air pollutants over oceans. *Atmos. Environ.* **1986**, *20*, 617–630. [CrossRef]
20. Muir, D.; Laxen, D.P.H. Black smoke as a surrogate for PM10 in health studies? *Atmos. Environ.* **1995**, *29*, 959–962. [CrossRef]
21. Smith, J.D.; Atkinson, D.B. A portable pulsed cavity ring-down transmissometer for measurement of the optical extinction of the atmospheric aerosol. *Analyst* **2001**, *126*, 1216–1220. [CrossRef]
22. Hodgeson, J.A.; McClenny, W.A.; Hanst, P.L. Air Pollution Monitoring by Advanced Spectroscopic Techniques: A variety of spectroscopic methods are being used to detect air pollutants in the gas phase. *Science* **1973**, *182*, 248–258. [CrossRef]
23. Li, X.; Peng, L.; Hu, Y.; Shao, J.; Chi, T. Deep learning architecture for air quality predictions. *Environ. Sci. Pollut. Res.* **2016**, *23*, 22408–22417. [CrossRef]
24. Chen, J.; Chen, H.; Zheng, G.; Pan, J.Z.; Wu, H.; Zhang, N. Big smog meets web science: Smog disaster analysis based on social media and device data on the web. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Korea, 7–11 April 2014; pp. 505–510. [CrossRef]
25. Liu, Y.H.; Yu, Z.; Huang, Y.L.; Cai, M.; Xu, W.J.; Li, L. Characteristic analysis on uneven distribution of air pollution in cities. *Environ. Monit. China* **2011**, *27*, 93–96.
26. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
27. Liu, H.; Li, F.; Xu, F.; Lu, H. The evaluation of air quality using image quality. *Chin. J. Image Graph.* **2011**, *16*, 1030–1037.
28. Wang, H.; Yuan, X.; Wang, X.; Zhang, Y.; Dai, Q. Real-time air quality estimation based on color image processing. In Proceedings of the 2014 IEEE Visual Communications and Image Processing Conference, Valletta, Malta, 7–10 December 2014; pp. 326–329.
29. Zhang, Z.; Ma, H.; Fu, H.; Wang, X. Outdoor air quality inference from single image. In Proceedings of the International Conference on Multimedia Modeling, Sydney, Australia, 5–7 January 2015; pp. 13–25.
30. Zhang, C.; Yan, J.; Li, C.; Rui, X.; Liu, L.; Bie, R. On estimating air pollution from photos using convolutional neural network. In Proceedings of the 24th ACM international conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 297–301.
31. Chakma, A.; Vizena, B.; Cao, T.; Lin, J.; Zhang, J. Image-based air quality analysis using deep convolutional neural network. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3949–3952.
32. Xing, H.; Wang, G.; Liu, C.; Suo, M. PM2.5 concentration modeling and prediction by using temperature-based deep belief network. *Neural Netw.* **2021**, *133*, 157–165. [CrossRef] [PubMed]
33. Song, Y.-Z.; Yang, H.-L.; Peng, J.-H.; Song, Y.-R.; Sun, Q.; Li, Y. Estimating PM2.5 concentrations in Xi'an City using a generalized additive model with multi-source monitoring data. *PLoS ONE* **2015**, *10*, e0142149. [CrossRef]
34. Sentinel Sentinel-Hub. Available online: https://apps.sentinel-hub.com/ (accessed on 13 February 2022).
35. AirNow Air Quality Data. Available online: https://www.airnow.gov/ (accessed on 22 February 2022).
36. LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. Handwritten digit recognition with a back-propagation network. *Adv. Neural Inf. Process. Syst.* **1989**, *2*, 1–4.
37. Vaillant, R.; Monrocq, C.; Le Cun, Y. Original approach for the localisation of objects in images. *IEE Proc.-Vis. Image Signal Process* **1994**, *141*, 245–250. [CrossRef]
38. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
39. Nowlan, S.J.; Platt, J.C. A convolutional neural network hand tracker. *Adv. Neural Inf. Process. Syst.* **1995**, *1*, 901–908.
40. Lawrence, S.; Giles, C.L.; Tsoi, A.C.; Back, A.D. Face recognition: A convolutional neural-network approach. *IEEE Trans. Neural Netw.* **1997**, *8*, 98–113. [CrossRef]

41. Hariharan, B.; Arbelaez, P.; Girshick, R.; Malik, J. Object instance segmentation and fine-grained localization using hypercolumns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 627–639. [CrossRef]
42. Garcia, C.; Delakis, M. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1408–1423. [CrossRef] [PubMed]
43. Riesenhuber, M.; Poggio, T. Hierarchical models of object recognition in cortex. *Nat. Neurosci.* **1999**, *2*, 1019–1025. [CrossRef] [PubMed]
44. Ciregan, D.; Meier, U.; Schmidhuber, J. Multi-column deep neural networks for image classification. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3642–3649.
45. Yao, S.; Xu, Y.-P.; Ramezani, E. Optimal long-term prediction of Taiwan's transport energy by convolutional neural network and wildebeest herd optimizer. *Energy Rep.* **2021**, *7*, 218–227. [CrossRef]
46. Fahlgren, N.; Feldman, M.; Gehan, M.A.; Wilson, M.S.; Shyu, C.; Bryant, D.W.; Hill, S.T.; McEntee, C.J.; Warnasooriya, S.N.; Kumar, I. A versatile phenotyping system and analytics platform reveals diverse temporal responses to water availability in Setaria. *Mol. Plant* **2015**, *8*, 1520–1535. [CrossRef] [PubMed]
47. Li, Y.; Huang, J.; Luo, J. Using user generated online photos to estimate and monitor air pollution in major cities. In Proceedings of the 7th International Conference on Internet Multimedia Computing and Service, Zhangjiajie, China, 19–21 August 2015; pp. 1–5.
48. Zhang, Q.; Fu, F.; Tian, R. A deep learning and image-based model for air quality estimation. *Sci. Total Environ.* **2020**, *724*, 138178. [CrossRef] [PubMed]
49. Wang, Q.; Hao, D.; Li, F.; Guan, X.; Chen, P. Development of a new framework to identify pathways from socioeconomic development to environmental pollution. *J. Clean. Prod.* **2020**, *253*, 119962. [CrossRef]
50. Pakistan, U. Available online: https://www.pk.undp.org/content/pakistan/en/home/library/development_policy/dap-vol7-issue2-environmental-sustainability-in-pakistan.html (accessed on 11 January 2022).
51. Rana, I.A.; Bhatti, S.S. Lahore, Pakistan–Urbanization challenges and opportunities. *Cities* **2018**, *72*, 348–355. [CrossRef]
52. Raziq, A.; Xu, A.; Li, Y.; Zhao, Q. Monitoring of land use/land cover changes and urban sprawl in Peshawar City in Khyber Pakhtunkhwa: An application of geo-information techniques using of multi-temporal satellite data. *J. Remote Sens. GIS* **2016**, *5*, 174. [CrossRef]
53. Mehmood, R.; Mehmood, S.A.; Butt, M.A.; Younas, I.; Adrees, M. Spatiotemporal analysis of urban sprawl and its contributions to climate and environment of Peshawar using remote sensing and GIS techniques. *J. Geogr. Inf. Syst.* **2016**, *8*, 137–148. [CrossRef]
54. Shupler, M.; Godwin, W.; Frostad, J.; Gustafson, P.; Arku, R.E.; Brauer, M. Global estimation of exposure to fine particulate matter (PM2.5) from household air pollution. *Environ. Int.* **2018**, *120*, 354–363. [CrossRef]
55. IQAir Air Quality in Lahore. Available online: https://www.iqair.com/pakistan/punjab/lahore (accessed on 15 January 2022).
56. IQair IQAIR. Available online: https://www.iqair.com/world-air-quality-ranking (accessed on 3 January 2022).
57. Pudasaini, B.; Kanaparthi, M.; Scrimgeour, J.; Banerjee, N.; Mondal, S.; Skufca, J.; Dhaniyala, S. Estimating PM2.5 from photographs. *Atmos. Environ. X* **2020**, *5*, 100063. [CrossRef]

remote sensing

MDPI

Article

Forecast of the Global TEC by Nearest Neighbour Technique

Enric Monte-Moreno [1,*,†], Heng Yang [2,3,†] and Manuel Hernández-Pajares [3,4,†]

1 Department of TSC, TALP, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain
2 School of Electronic Information and Engineering, Yangtze Normal University, Chongqing 408100, China; h.yang@upc.edu
3 Department of Mathematics, UPC-IonSAT, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain; manuel.hernandez@upc.edu
4 Institut d'Estudis Espacials de Catalunya (IEEC), 08034 Barcelona, Spain
* Correspondence: enric.monte@upc.edu
† These authors contributed equally to this work.

Abstract: We propose a method for Global Ionospheric Maps of Total Electron Content forecasting using the Nearest Neighbour method. The assumption is that in a database of global ionosphere maps spanning more than two solar cycles, one can select a set of past observations that have similar geomagnetic conditions to those of the current map. The assumption is that the current ionospheric condition can be expressed by a linear combination of conditions seen in the past. The average of these maps leads to common geomagnetic components being preserved and those not shared by several maps being reduced. The method is based on searching the historical database for the dates of the maps closest to the current map and using as a prediction the maps in the database that correspond to time shifts on the prediction horizons. In contrast to other methods of machine learning, the implementation only requires a distance computation and does not need a previous step of model training and adjustment for each prediction horizon. It also provides confidence intervals for the forecast. The method has been analyzed for two full years (2015 and 2018), for selected days of 2015 and 2018, i.e., two storm days and two non-storm days and the performance of the system has been compared with CODE (24- and 48-h forecast horizons).

Keywords: Global Ionospheric Maps; Total Electron Content forecasting; machine learning; Nearest Neighbour method

Citation: Monte-Moreno, E.; Yang, H.; Hernández-Pajares, M. Forecast of the Global TEC by Nearest Neighbour Technique. *Remote Sens.* **2022**, *14*, 1361. https://doi.org/10.3390/rs14061361

Academic Editor: Michael E. Gorbunov

Received: 19 January 2022
Accepted: 9 March 2022
Published: 11 March 2022

1. Introduction

The variations in electron density and, correspondingly in its line-of-sight integral, the vertical total ionospheric electron content (TEC), affect satellite telecommunication services and Global Navigation Satellite Systems (GNSS) due to the effect these fluctuations have on radio wave propagation. The TEC variations induce changes that affect the transmission quality either as reduced transmission rate or positioning errors. This justifies the importance of monitoring and predicting global TEC maps, as the knowledge of the spatial distribution of TEC would allow corrections to be made. The TEC measurement consists of the total number of electrons integrated along a 1 m^2 cross-section tube, using as a unit the TECU defined as = 10^{16} electrons/m^2. The prediction of Global Ionospheric Maps (GIM) at different horizons is important because the ionospheric delay is main limiting factor in high-accuracy positioning. These predictions may allow achieving sub-meter accuracy for mass-market single-frequency receivers [1]. In this paper we propose a method for Global Ionospheric Maps of Total Electron Content forecasting using the nearest neighbour method which we denote as NNGIM.

1.1. Issues Related to Previous Work in TEC Map Prediction

The difficulty in predicting TEC maps of the ionosphere stems from the fact that the quality of the prediction depends on geomagnetic activity, season, geographical location, ionospheric structures, such as equatorial ionization anomaly (EIA), and storm-enhanced density (SED). Besides, the sparsity in the geographical distribution of stations leads to problems related to interpolation in regions not covered by these stations. Added to the problem of variability and dependence on external factors, the prediction of GIM maps by machine learning techniques is affected by the need for machine learning techniques to infer prediction rules from examples. This means that the database to train the system has to be rich enough to represent most of the combinations of effects acting on the ionosphere. One intrinsic limitation of machine learning-based systems is the availability of a database that sufficiently covers the multiple forms of phenomena that can occur. In the works cited below, most of the prediction proposals are made using databases covering at most one solar cycle. In this work, we will be using UPC-IonSAT's database (for more information about the IonSAT group, i.e., ionospheric determination and navigation based on satellite and terrestrial systems group see [2]), which covers more than two solar cycles. It is important to highlight the importance of having more than one solar cycle to infer the structure and parameters of the forecasting system. Within the long-term solar cycle periodicity, there is large variability. As an example analyzed in this paper, we can mention two dates when storms occur. For instance, the Saint Patrick storm of 17 March 2015 (maximum of solar cycle C23) and the storm of 25–26 August 2018 (minimum of solar cycle C23). These are dates in different phases of the solar cycle, in which we have high solar and geomagnetic activity superimposed on different basal levels of ionization. The Tables 1 and 2, summarise the hourly Kp values on these days. In these two days, the activity in terms of Kp values and magnitude of the flares is similar. Therefore, within the periodicity associated with the solar cycles and the season of the year, there is a high variability that makes it difficult to infer prediction rules. This high variability, in addition to the baseline levels of activity due to the periodicity components, justifies the need for a long enough database.

Table 1. Hourly Kp for the 17 March 2015.

Hour	00–03 h	03–06 h	06–09 h	09–12 h	12–15 h	15-18 h	18–21 h	21–00 h
Kp (17 March)	2	5	6	6	8	8	7	8

Table 2. Hourly Kp for the 25–26 August 2018.

Hour	00–03 h	03–06 h	06–09 h	09–12 h	12–15 h	15–18 h	18–21 h	21–00 h
Kp (25 August)	1	1	2	2	3	2	4	4
Kp (26 August)	5	7	7	5	5	6	5	3

The need for a database that sufficiently covers the variability of GIMs presents significant technical problems from the point of view of prediction algorithms. In the case of two solar cycles, with maps at a rate of one every 15 min, the resulting database consists of more than one million maps. The use of databases of this size makes the hardware requirements demanding, and the computational time requirements to perform topology and parameter tuning of the machine learning system are substantial.

To address the above mentioned problem, i.e., of training a machine learning system for forecasting the GIMs, making, there are two approaches.

Local approach: In this case, a specific subset of the database is constructed from the current observation. An example is [3], in which maps immediately before the current map are used, and the forecasting method is based on these maps and the associated tangent spaces, which are linearly combined to generate the predicted maps. This approach assumes that the change in the maps has inertia that determines the future evolution. In [4], they apply a similar idea to calculate the autoregression coefficients that predict the values of the spherical harmonics that allow the GIMs to be reconstructed. Another approach is

the one followed in this article, in which prediction is made based on past examples that have a small distance to the current observation. This approach assumes that conditions similar to the one observed in the current map have occurred in the near past and that the temporal evolution of the current map can be inferred from the evolutions seen in the previous history. A noteworthy aspect of the local approximation is that increasing the number of prediction horizons does not lead to a significant increase in computation time, as most of the computation time comes from determining the coefficients in a window that spans a limited amount of time.

Global approach: In this case, the prediction model uses all the historical GIMs. One consequence of this is that to make a reliable prediction, the model has to be estimated from a sufficiently rich set of examples. This leads to problems of implementation. For support vector machines, this approach is infeasible, since it is necessary to create the gram matrix, which is the square of the number of examples, and it must be kept in memory. In the case of deep learning [5], the training has to be carried out in graphical processing units (GPU), which have limited memory.

Another significant limitation in the approach using deep learning and similar methods is that either a completely new model or a more complicated topology has to be trained when increasing the number of prediction horizons. In contrast, the method we propose, the nearest neighbour GIM algorithm (NNGIM), is based on finding the nearest set of maps, increasing or changing the values of the horizons has minimal repercussions on the execution time.

A natural model for forecasting the GIM maps that has been used in the literature (see Section 1.2) is the long short term memory (LSTM) [5] architecture. A very significant limitation of the LSTM architectures is that they consist of units that have saturating nonlinearities, such as hyperbolic tangent and sigmoid. Since the GIM statistics are long tail (see the last section of [3]), the units work much of the time in saturation and cannot model large amplitudes. For a complete explanation see Section 4.4. One consequence is that precisely the regions of interest where there are large TEC gradients cannot be modelled correctly by these units, as the gradient is zero due to the saturation of the nonlinearities. The complexity of using Deep Learning based methods was one of the motivations for seeking a more simple approach to the problem.

1.2. Approaches and Limitations to the GIM Forecast

We will now discuss some antecedents to set the NNGIM in context. The features and limitations of other GIM prediction methods will allow us to justify NNGIM design decisions. This section will also serve to highlight the limitations of the global approach to forecasting.

Global approach: A first approach to the problem of predicting TEC maps consists of predicting TEC values for specific stations, thus obtaining a local description of the TEC distribution. This is the case of [6], where they predict the TEC over China using a variant of the LSTM type networks (ED-LSTM). This type of method differs from ours in the sense that the prediction is done at the station level and there is no interpolation process. One point to note is the use of data from one solar cycle (January 2006 to April 2018). The authors use training data from 2006 to 2016, validation between January 2017 and April 2018. To avoid the problem of the solar cycle-dependent baseline TEC level, and to adapt the data to the structure of the LSTM grids, the authors normalise the data. This assumes that the variations around the baseline TEC value are similar between different times of the solar cycle. One problem related to their approach is that the neural network units they apply have saturation-type non-linearities, which has as a consequence that for extreme values, the units work on saturation (i.e., the gradients are null, and the extreme value is set to the limit of the saturation function). Note that the statistics of the TEC distribution is Leptokurtic, i.e., long tail, which means that extreme values are much more common than expected in the case of a Gaussian or Exponential distribution. On the other hand, an advantage of the type of neural network they employ is that it allows the use of external

data naturally in the architecture (solar flux and geomagnetic activity data). In addition to the LSTM architecture (ED-LSTM), the authors explore other architectures and provide a performance hierarchy. The forecast horizons are 2-h, 3-h, and 4-h, using as input a window of past samples between one day and three days. An important lesson from this work is that the inertia hypothesis, in the sense that the temporal evolution of the TEC follows a trajectory specified by the near past, leads to a prediction barrier at a horizon of a few hours. This limit on the prediction horizon under these conditions was also found in [3].

Another paper working on the prediction of TEC values using a network of local stations in Turkey is [7]. Unlike the present paper, which deals with stations distributed all over the globe, the authors use five stations, all located in the mid-latitude region. The training implementation uses inputs corresponding to the current TEC value, together with measurements affecting the evolution of ionization, such as Kp, solar flux (F10.7 cm), magnetic field (Bx, By and Bz) and proton density, and EUV radiation in two bands. The neural network they use is based on LSTM structures, which suffer from the above-mentioned drawback, i.e., the input signal has a Leptokurtic statistic. In other words, outliers are common (for instance Figures 5 to 7 of the article [6]), while the prediction mechanism is based on LSTM units that saturate at high levels of any of the inputs. This means that in situations of high ionization variations, this approach does not allow the prediction model to learn from these variations. Another aspect that concerns us in the current implementation is the robustness of the prediction system with respect to the measurements. The fact of using heterogeneous measurements as input to the network makes the prediction susceptible to events of loss of some type of measurement.

A similar paper is [8] that performs the map prediction on a single meridian, 120 degrees, in a range of latitudes between 80 degrees north and 80 degrees south, (in contrast to our case, where we perform a global prediction). They use as input to the system a history of past measurements of daily TEC sampled at 2-h intervals, together with the mean value of the solar flux. An interesting feature of this work is that the use of external information (kp and Dst) had a different influence depending on the phase of the solar cycle. Another limitation, which is common to other applications using neural networks, is the partitioning between train, test and validation. In this case, for the validation partition, the years 2015 and 2018 were used, which correspond to the time after the peak of activity and when the activity decreases. Since the statistics of the TEC variation and the external information used are different according to the phase of the solar cycle, this partition introduces a bias in the architecture and parameters selected for the predictor. Moreover, the use of sigmoid/hyperbolic nonlinearities in LSTM/MLP prediction methods leads to the limitations discussed in Section 4.4.

An article reporting a related architecture is [9]. Unlike the previous case, the objective was to predict global TEC maps, with a resolution of 5 by 2.5 degrees in longitude and latitude. The temporal resolution was 2 h. To solve the diurnal cyclicity problem, they use a solar centred reference frame. The authors propose the prediction of global maps with prediction horizons increasing in two-hour steps up to 48 h. The input data were the maps for the three immediately preceding days. The type of architecture they propose is based on a sequence to sequence, in which CNN-type networks are combined with memory networks, either LSTM or gated recurrent units (GRU), both with saturating nonlinearities. The authors report that prediction at intervals longer than 24 h did not achieve good results; in fact, in the 24-h prediction, they obtain a result that improves the cyclic prediction by only 6%. The study was conducted using the data from 1 January 2014 to 31 December 2016. Note also, that the use of LSTM or GRU also suffers from the limitation that the observations are leptokurtic, which means that the nolinearities work in saturation for extreme values.

In [10] the authors propose a system based on the use of two LSTM layers followed by a fully connected dense layer for the prediction of the global TEC maps. Unlike the previous cases, the prediction is performed directly on the spherical harmonic (SH) used to build the GIMs. In this approach, in addition to using the information in the recent past (24 h)

regarding the SH, they also use external information that helps to make the prediction, such as the solar extreme ultraviolet (EUV) flux, the hour of the day, and disturbance storm time (Dst) index. The prediction horizon is set to 1 h and 2 h. It is interesting to note that the prediction has an error with respect to frozen maps (defined as persistence, i.e., $\hat{Map}_{frozen}(t+\tau) = Map(t)$) of 60% at one hour and 63% at two hours. Note that (although the experiment is not totally comparable) this gain is similar to the obtained by the frozen cyclic approach vs. the persistence hypothesis, in Section 2.5 of the current paper. As a test base, the intervals before and after the interval used for the training base were used. That is, for the training base the interval: 1 January 2015 to 26 May 2016 and for the test base the intervals 19 October to 31 December 2014 and 27 May to 31 December 2016, thus ensuring a similarity between the training and test conditions.

The methodology of the above-mentioned works is correct from the point of view of deep learning type network design, however, despite the correctness, it reflects the limitations of this type of technique. These limitations are typical of the general approach to the TEC prediction problem using deep learning and do not indicate a misuse of the technique by the authors. The limitations of Deep Learning are the need to process the input data such as normalisation or de-trending of the TEC, the difficulty of performing a test under train-like conditions, the fact that some networks require saturating nonlinearities that are not fit for long-tail input distributions, and the limitations for predictions at horizons greater than 24 h.

A different approach to the GIMs prediction problem is the one proposed by [11], employing GANs (Generalized Adversarial Networks), which consists of a generative method, with a training criterion based on generating maps that compete with a system that generates impostors. It is a method that, by observing the current GIM map, generates the future one. Unlike most prediction systems, it does not depend on a previous history of GIM measurements, so it is robust to loss/reinitialization of the GIM data source. Like our method, it implicitly assumes that the external conditions that determine the evolution of the maps are included in the current measurement. However, an important limitation of this method is that the quality depends on the data used for training and validation. In the case of this publication this limitation is crucial, because the partition that was performed to train the method ((1) a training data set (2001–2011), (2) a validation data set (2012), and (3) a test data set (2013–2017)) makes that the criteria to determine the characteristics of the experiment given by the year of validation, induce a bias that makes the behavior of the predictions on unseen data depend on the accidental conditions of this partition and the peculiarities of the chosen cycle. This method does not have the limitations of the above mentioned methods in the sense that it does not use nonlinearities with saturation, and does not depend on additional measures to the GIM, which makes it robust to data loss.

Local approach: This approach uses information from recent past to estimate the parameters of the prediction model.

In [4], the authors describe a system based on autoregressive models, with coefficients computed from a history covering the previous 30 days. The prediction is made on the SH coefficients, which allow the GIM to be reconstructed. By estimating the model locally, they can adapt the system to short-term climatology. This allows them to test the model at different times of the solar cycle, without the need for special partitioning of the database, as is done in the case of deep learning. The performance of the model is tested against CODE, IGS products, and TEC measurements via JASON. The prediction result is different depending on the activity at the time, with worse results at times of high activity. One result is that the RMSE error of prediction during a low activity period was 1.5 TECUs at 24 h. In [12] the authors use autoregressive moving average (ARMA) for vertical TEC (VTEC) prediction for stations in Northern Europe. In this article, they use information related to the analysis in wavelets to establish the prediction at 1, 2, and 3-h horizons, calculating the ARMA coefficients from the last 7 days. The TEC profiles follow a daily pattern, so an ARMA-type method is suitable for modeling the cyclicities.

In [1], the authors propose a method for the prediction of GIMs with horizons of up to 2 days. It is based on a method that predicts the coefficients of the discrete cosine transform (DCT) by an autoregressive method. The autoregressive coefficients are calculated locally using information from the last week's maps. From the predicted DCT coefficients, the map at the horizon of interest is computed. By calculating the coefficients using a recent past and using the maps of the previous 24 h for the prediction, the system can adapt to the current weather conditions. The results were validated with JASON measurements.

In [3] a prediction system is proposed based on an autoregressive model of the maps of the last 24 h, updated using only recent observations. The forecast also uses the components of the tangent spaces associated with each of the previous maps. The forecast horizons range from half an hour to 24 h. The tangent space information allows an increase in the information on the possible trajectory and deformation of the map over time, and in some way to reflect how the ionospheric climatology changes the shape of the high ionization regions. One feature related to the comparison with other methods is the percentage improvement of the prediction method compared to a frozen reference in a sun-fixed reference frame. The reference will be the prediction error of keeping the map frozen (see Section 2.5 for more information). As shown in Table 3, the prediction performance has a concave profile. The performance is computed using the recent past, and with autoregressive model coefficients calculated with recent values as well. The best prediction compared to frozen is at a 3-h horizon, increasing thereafter. At 24 h, the improvement is only 5%, which is in line with methods based on deep learning. This leads us to think that there is a certain horizon barrier in terms of prediction using the recent past as input.

Table 3. Forecast vs. frozen (% RMSE) for the tangent space.

Horizon:	**1/2 h**	**1 h**	**2 h**	**3 h**	**6 h**	**24 h**
Forcast vs. Frozen:	84.99%	77.65%	71.35%	69.34%	87.23%	95.76%

The analysis of the previous approaches leads us to the conclusion that the information immediately prior to the current map does not allow reliable predictions of GIM maps at horizons longer than a few hours. They also indicate the limitations and difficulties of training prediction models, and the complexity of the models and partitions of the database.

This leads us to look for a different approach, in which the prediction is made by searching for situations similar to the current one in a sufficiently large database. A by-product of this approach is that it allows the creation of confidence margins of the forecast in a natural way (see Section 4.1).

2. Materials and Methods

2.1. UPC-IonSAT Real-Time Global Ionospheric Maps and Data Preprocessing

The GIMs are generated from data gathered from several hundred worldwide GNSS stations. This data stream is obtained through the protocol used by the RT IGS working group and the data processing is performed using the UPC-IonSAT ionosphere model.

The streaming protocol referred to as "Networked Transport of Radio Technical Commission for Maritime Services (RTCM) via Internet Protocol" (NTRIP), was developed by the German Federal Agency for Cartography and Geodesy (BKG), enables the streaming of the observation data from the worldwide permanent GNSS receivers [13].

The UPC-IonSAT's RT TOMographic IONosphere Model (RT-TOMION) is a 4D (3D+time) model of the global state of the ionosphere, focused on RT estimation of TEC, mainly based on GPS dual-frequency measurements with the hybrid geodetic and tomographic ionospheric model, and robust to various types of deterioration. This model is the extension of the Tomographic Ionospheric Model (TOMION) developed by UPC in the 1990s and has been employed for UPC RT/near-RT ionosphere service of IGS since 2011 [14–18].

Additionally, the VTEC interpolation techniques of the UPC RT- TOMION model are performed either by spherical harmonics or Kriging [16] so to fill the gaps where data is lacking. In addition, the most recent maps are interpolated by means of the ADDGIM algorithm presented in [19]. For more details of the processing and interpolation of the GIMs, see [19].

2.2. The NNGIM Forecasting Algorithm

In this section, we will describe the Nearest Neighbour GIM (NNGIM) algorithm. This algorithm consists of searching for the N maps closest (in Euclidean metric) to the current one in the database of past maps (more than one solar cycle). Then, from these maps, the GIMs with an offset equal to the prediction horizon are retrieved and averaged.

The assumption underlying the NNGIM algorithm is that in a database that encompasses more than one solar cycle, a small number of maps with the property of being the closest in Euclidean distance to the current one can be found, and that have ionosphere conditions in common with the current one, might characterize the maps at a time shift equal to the forecast horizon. Although each ionosphere condition is unique, it is assumed that in the past there have been conditions with a similar composition of external features and that the average of all of them will reflect the specific features of the current one. The set of similar maps therefore take into account the cyclical aspects that influence the overall distribution of TEC along with the various external influences. That is, if we select a set of future map values closer to the current one when averaging, common values in subsets of the future maps will be retained, while non-common conditions will be attenuated. Note that the idea behind the assumption is that there will be subsets of maps representing similar ionospheric conditions, and the overall composition of these parts will allow us to approximate previously unseen situations. We assume that these previously unseen situations are composed of subgroups that characterize part of the previous conditions common to the current situation.

The UPC-IonSat GIMs database, which spans over two solar cycles and consists of more than 10^6 maps, was used to implement the method (see [19] for details).

In the algorithm diagram, Algorithm 1, we present the summary of the NNGIM algorithm. A detailed explanation of the algorithm is given below, also defining the variables involved.

The input of the algorithm consists of a database spanning more than two solar cycles ($Db_{AllMaps}$). Note that for consistency in the computation of the distance between maps at different moments, the database and the current map are transformed to sun-fixed geomagnetic coordinates. After the forecast, the inverse transform is performed.

Since the maps have a seasonal component with a mean TEC value that depends on the season of the year (see Figure 1), the search for the nearest map will be carried out in the vicinity of the current month. Therefore, given the date of the current map $Date_{Test}$, the month is extracted (M_{Tst}), and maps the current month and a window of $\pm W_{NeighMonths}$ months are selected from the database. In the experiments, a neighbourhood of $W_{NeighMonths} = 1$ was taken. Other parameters are the forecast horizon in hours ($Horizon$) and the number of nearest neighbours (Num_{NN}). The next step is to construct a second database (Db_{Ima}), which will consist of the maps with the current map month and the neighbouring months for all years. The Euclidean distance between the current map $Map(Date_{Test})$ and the maps in the Db_{Ima} database is then calculated (lines 3 to 7 of the Algorithm 1). The vector of distances is then sorted from smallest to largest (line 8 of the Algorithm 1) and assigned to the vector of indices $Index_{MinDist}$.

We define Num_{NN} as the number of maps to be used for prediction estimation. The Algorithm 1, lines 9 to 15 describe the process for generating the prediction. For the nearest Num_{NN} maps, we find the corresponding index $IndexMap$ and the associated date $Date[IndexMap]$. Next, we add the offset $Horizon$ to generate the date $Date_{NNMap}$ associated with each of the maps. The maps associated with each date $Date_{FutMap} \leftarrow Date_{NNMap} + Horizon$ are combined to generate the future map $Forecast_{Map}$.

Finally, from the maps of the horizon shift, the standard deviation at the pixel level is calculated, as shown in line 17.

Algorithm 1: The NNGIM algorithm.

Data: Inputs to the algorithm:
$Date_{Test}$ ← Date of the test GIM;
$Db_{AllMaps}$ ← All GIMs of two solar cycles in sun-fixed geomagnetic coordinates;
W_{Neigh} ← Window of Neighbouring Months;
Num_{NN} ← Number of elements for computing the mean of the Nearest Neighbours;
$Date$ ← Dictionary of Dates, indexed by Map number;
$Horizon$ ← Forecast Horizon in hours;
Result: $Forecast_{Map}$,$Forecast^{Std}_{Map}$

1 **Generate the Forecast Database**;
2 $M_{Tst} \leftarrow GetMonth(Date_{Test})$; /* Month of the current map */
3 $Db_{Ima} \leftarrow \emptyset$; /* Db_{Ima} Map DataBase of Current and Neighbouring Months */
4 **for** M=$M_{Tst} - W_{Neigh}$ *to* $M_{Tst} + W_{Neigh}$ **do**
5 | $Db_{Ima} \leftarrow (Add\ to\ set) Db_{AllMaps}(M)$; /* Add maps for month M */
6 **end**
7 $Mat_{Dist} \leftarrow Distance(Db_{Ima}, Map(Date_{Test}))$; /* Distance from $Map(Date_{Test})$ to Mat_{Dist} */
8 $Index_{MinDist} = Argsort(Mat_{Dist})$; /* Argsort returns the Indices of the sorted Mat_{Dist} */
9 $For_{Map} \leftarrow \emptyset$; /* Compute mean value of the nearest maps at timestamp + horizon */
10 **for** $NumMap$=1 ***to*** Num_{NN} **do**
11 | $IndexMap \leftarrow Index_{MinDist}[NumMap]$;
12 | $Date_{NNMap} \leftarrow Date[IndexMap]$;
13 | $Date_{FutMap} \leftarrow Date_{NNMap} + Horizon$;
14 | $For_{Map} \leftarrow For_{Map} + Db_{AllMaps}[Date_{FutMap}]$;
15 **end**
16 $Forecast_{Map} \leftarrow For_{Map} / Num_{NN}$;
17 $Forecast^{Std}_{Map} \leftarrow Compute_{STD}(Db_{AllMaps}, Date, Index_{MinDist}, Horizon)$;

Various strategies for combining the maps were tested, such as a simple average, a distance-weighted average, or weight that diminishes with the time difference. We also tried a trim mean, defined as the average of the values of each specific pixel in the maps, using only the values between the 25th percentile and the 75th percentile. The median of the pixels of the nearest Num_{NN} maps was also tested. The combination that gave the best results was a simple average of the maps.

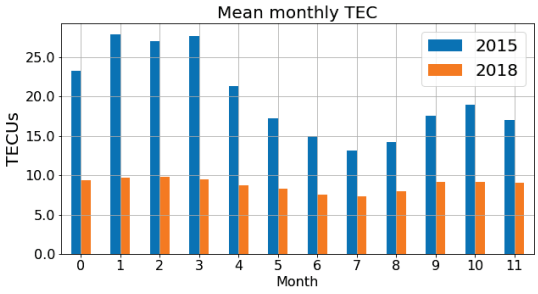


Figure 1. Mean monthly TEC for the years 2015 (in blue) and 2018 (in orange).

One parameter to be adjusted is the number Num_{NN} used to calculate the forecast. This value depends on the forecast horizon and the month of the year. For all experiments we chose a value Num_{NN} = 500. The choice was made based on the performance during June 2019 and was explored for values between 1 and 1000. The rationale for the choice of date was to have a date in a cycle (C24) different from the cycle in which the results are presented (C23), and also at a season of low activity. The experiments showed that for this month and horizons between 3 h and 48 h the optimum value was between 150 and 700. In the real-time implementation, a look-up table will be used in which the month and horizon will be related to the Num_{NN} value.

An interesting result is that using only the nearest neighbour, i.e., $Num_{NN} = 1$ provided results with a quality equal to using the cyclic version of the map, (defined as $\hat{M}ap_{cyclic}(t + \tau) = Map(t - 24h + \tau)$). The performance did not improve until using a number of Num_{NN} greater than 50. This leads us to think that the use of a large number of maps allows us to create a representation of the possible contributions of the factors that affect ionisation. The explanation is that the combination of external factors is larger than the number of examples in the database. The underlying assumption is that the current combination of factors affecting ionisation can be expressed as a linear combination of similar situations in the past.

A product of this algorithm is that it can provide confidence intervals for the GIMs, i.e., the local standard deviation of the ionisation values. The estimation of confidence intervals can be done directly, as a collection of several hundred maps is available. One of the features of the maps from which the prediction is constructed is the variability around a central value, as shown in Figure 2. Therefore from the set of maps used to generate the prediction, one can estimate a standard deviation $Forecast^{Std}_{Map}$ at a pixel level, defining this standard deviation as the deviation of the maps from the mean value of the prediction $Forecast_{Map}$. One point that we show in Section 4.1 is that the prediction covers most of the area of the reference map Ref_{Map}, so we can consider that this variance provides us with an adequate measure of uncertainty for the prediction.

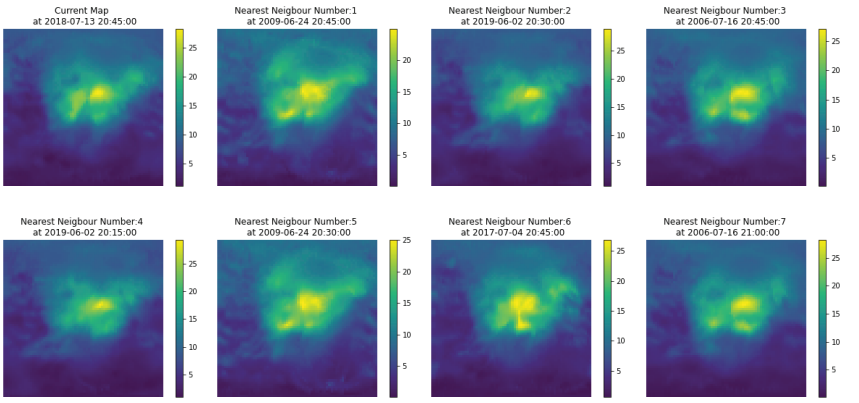


Figure 2. Current map at *2018-07-13 20:45:00 UTC* (subplot at upper left corner), and the seven Nearest Neighbours. All maps in sun-fixed geomagnetic coordinates. The maps range in latitude from 90 degrees north to 90 degrees south, and in longitude from 180 degrees west to 180 degrees east. Color bars are in TEC units.

Parameter setting. The algorithm has two parameters to be adjusted, which are the window in months to select maps, denoted as W_{Neigh} and the number of elements to calculate the mean value of the nearest neighbors, denoted as Num_{NN}. The criterion to adjust the parameters was to fit on a subset of the training base (the test was not used at any time for adjustment). In the case of W_{Neigh}, which corresponds to the neighboring months, it was observed that due to the variation of the algorithm itself, examples were always selected either from the current month or from the neighboring months. In order to limit the calculation needs, the calculation of distances was limited to the intervals determined by this variable. As for the Num_{NN} variable, the result is different from the normal application of the Nearest Neighbour (NN) algorithm, in the sense that in order to compensate for the specific variability of each example used for the prediction, the number of neighbors to be used is much higher than in normal applications of the NN. In our case, the prediction error decreased monotonically until reaching a NumNNN value of about 500, producing a plateau of error with small oscillations of the error until reaching about 1500, and at this point the error starts to increase. Note that the fraction of elements used is small with respect to the total number of examples which exceeds one million.

To see the effects of adjusting these parameters, see illustration of how the algorithm works (Section 2.3) and example of forecasts at several horizons (Section 2.4).

Improvements: The improvements we envisage in the next step are to change the average distance, using a metric on the manifold in which the map is located. This is the distance defined in [20] in which coefficients of the angle between coordinates $g_{i,j} =< e_i, e_j >$ are used to weight the Euclidean distance. The advantage of using this distance is that it allows considering in the similarity measure between maps, distortions such as shifts, rotations, etc. The reason why it has not been used in this implementation is that it requires a computational load proportional to the square of the number of map elements. With the current hardware capabilities at 2021, the computation of Mat_{Dist} took about ten minutes, so it was not implemented in the final prototype.

Another improvement is to use a heuristic that decreases the computational needs to determine the nearest neighbors. That is, an algorithm with a suitable heuristic for the dimensionality of the maps and with a lower search cost, as is the case of [21]. The fact that the GIMs have the ionisation levels distributed in clear and distinct regions makes this algorithm efficient. This might allow implementing a distance with higher computational cost as the nearest neighbour search cost can be decreased.

The computational cost on an iMac i7 using one core of applying the algorithm was as follows. The Euclidean distance Mat_{Dist} from a map $Map(Date_{Test})$ to the database Db_{Ima} consisting of the current month and the two neighbouring months (with 170,000 maps) was of the order of 135 ms, and the cost of sorting the distances $Argsort(Mat_{Dist})$ of 9 ms, the calculation of the average map $Forecast_{Map}$, was less than 1 ms.

The format of the maps consisted of TEC values measured with a resolution of 2.5 degrees in latitude and 5 degrees in longitude, resulting in maps represented as a 72×71 array of floats. Each map occupied 40 k bytes on disk (the float represented in ascii format had only one decimal place), while in memory it occupied 164 k bytes with a float-32 bit representation. For more information see Section 2.1

The most time-consuming part of the algorithm is the loading into memory of the pre-computed database Db_{Ima}, which occupies 2 gigabytes. The time cost on an SSD is in the order of 2 s. However, in a real-time application, the database can be kept permanently in memory.

The real-time prediction of the implementation of this algorithm can be found at the following URL: [22], with the following naming convention:

The three regions where the forecast was done: Global Forecast (un*g), North-Pole Forecast (un*n), South-Pole Forecast (un*s) And the different horizons that were implemented in real time:

1 un0g/un0n/un0s: 1 h Forecast
2 un1g/un1n/un1s: 6 h Forecast
3 un2g/un2n/un2s: 12 h Forecast
4 un3g/un3n/un3s: 18 h Forecast
5 un4g/un4n/un4s: 24 h Forecast
6 un8g/un8n/un8s: 48 h Forecast

The polar predictions consist of segments of the global map clipped at 45 degrees of latitude.

2.3. Illustration of How the Algorithm Works

To understand how the algorithm works, we will consider two points of view.

1. How the dates of the nearest maps are distributed along the solar cycles: *C23*, *C24* and *C25*.
2. Examples of actual maps to understand how is the variability of the nearest neighbours.

We will perform the analysis on day *2019-05-21 16:15:00 UTC* a *C25* cycle day during summer.

1. In Figure 3 we show that the nearest neighbours are distributed over years in the same phase of the cycle. Using only examples from the two cycles *C23* and *C24*. The algorithm does not select any maps from the previous month, and most of the closest maps are from the next month. As we will see later, there is a significant dependence of the behaviour of the algorithm on the month in which the prediction is made. As for the time of day, most of the examples are at the same time of day plus or minus one hour.
2. Next, we consider the variability of the closest maps. The variability of these maps reflects the ionospheric conditions that are common and those that differ. In Figure 2 we show the map for *2018-07-13 20:45:00 UTC* and the first seven nearest neighbours in the Euclidean distance sense. In the experiment we used 500 nearest neighbours to estimate the forecast. Examples of prediction are shown in Figures 4–6. To help make it easier to compare, we present the maps in sun-fixed geomagnetic coordinates, which are the setting in which the software computes the distance between maps. The selected maps are from the same time of the year and at similar moments of the solar cycle. On the other hand, the morphology is variable, which indicates that each of the maps reflects ionospheric conditions that have parts in common with the current

map as well as specific components. The hypothesis underlying the NNGIM model is that the components common to the current map are preserved by the average, and those that are not common are smoothed out. This variability around common values allows to estimate confidence intervals can capture the most likely ranges in the true reference value. The maps at a future shift equal to the prediction horizon exhibit very similar visual features. For reasons of space and similarity between figures, we do not show them.

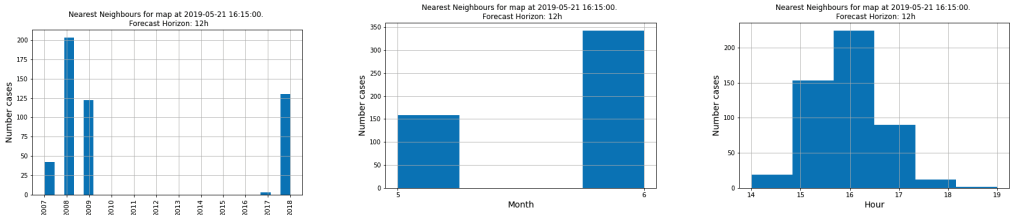


Figure 3. Nearest maps are distributed along solar cycles *C24* and *C25*. Histograms of the years (**left**), months (**center**) and time of day (**right**) of the nearest maps to the map at *2019-05-21 16:15:00 UTC*.

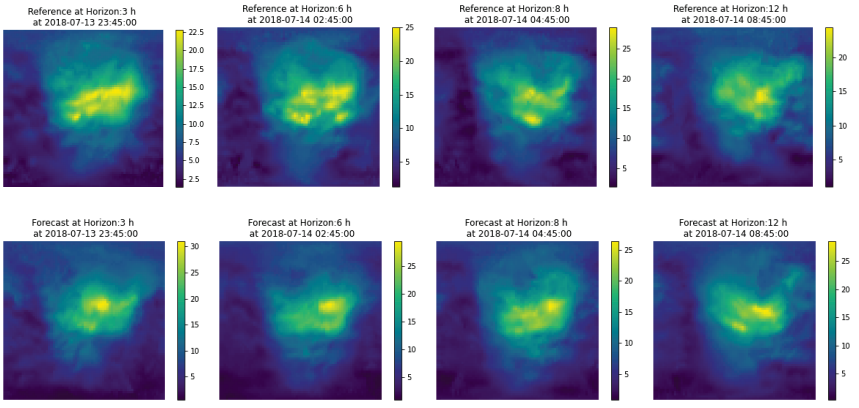


Figure 4. Selected sequence of predictions for the map at 2018-07-14 20:45:00 UT. The upper row shows the reference to 3 h, 6 h, 8 h, and 12 h horizons, the second row shows the prediction result. Note that the color bars are not at the same scale. The maps range in latitude from 90 degrees north to 90 degrees south, and in longitude from 180 degrees west to 180 degrees east. Color bars are in TEC units.

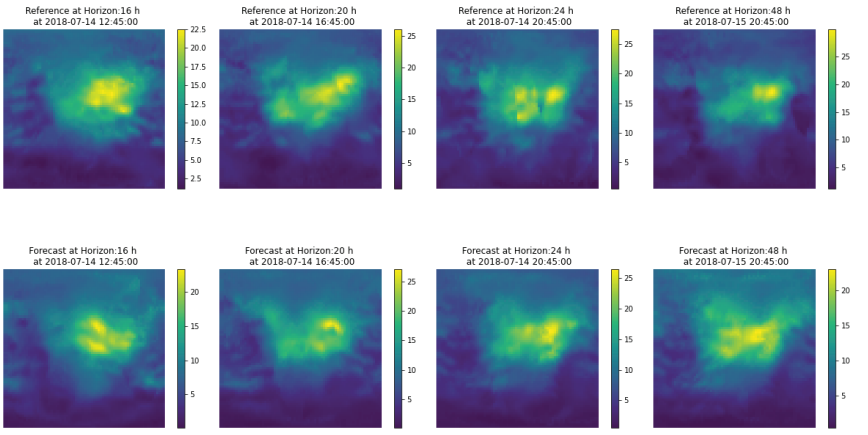
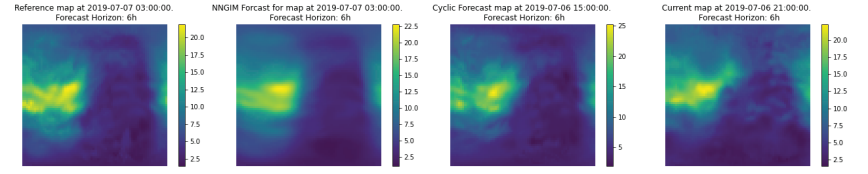


Figure 5. Selected sequence of predictions for the map at 2018-07-14 20:45:00 UT. The upper row shows the reference to 16 h, 20 h, 24, 48 h, the second row shows the prediction result. Note that the color bars are not at the same scale. The maps range in latitude from 90 degrees north to 90 degrees south, and in longitude from 180 degrees west to 180 degrees east. Color bars are in TEC units.



(**a**) $Map_{ref}(t+\tau)$ (**b**) $\hat{M}ap_{nngim}(t+\tau)$ (**c**) $\hat{M}ap_{cyclic}(t+\tau)$ (**d**) $\hat{M}ap_{frozen}(t+\tau)$

Figure 6. Comparison of the reference map (**a**) at *2019-07-07 03:00:00 UTC*, with the NNGIM prediction, (**b**), with the cyclic prediction, (**c**) and with the frozen prediction (i.e., using current map) (**d**) with the frozen map. Note that the maps are in the original coordinates, not in the sun-fixed geomagnetic coordinates.

2.4. Example of Forecasts at Several Horizons

In Figures 4 and 5 we show a selected sequence of predictions for the map at 2018-07-14 20:45:00 UT, at horizons ranging from 3 h to 48 h. In the first row we show the reference to 3 h, 6 h, 8 h, and 12 h horizons, and in the second row we show the prediction result. The third and fourth rows show the results for horizons of 16 h, 20 h, 24, 48 h. In order to assess the results it has to be taken into account that the colour bars are not at the same scale. This means that local maxima can distort the level of the overall colour gradation. In any case, an indication of the effectiveness of the algorithm lies in comparing the medium/high ionisation regions (not maxima) between reference and prediction. In these cases, the shape of the regions is found to be similar.

Note that the figures use as color code the 'viridis' scale instead of the more usual 'jet' scale. The reason is that the 'viridis' color scheme implements a linear scale with brightness going from dark black to bright yellow linearly, while the 'jet' scale has the brighter colors at the middle of the scale (blue/yellow), and the lowest/highest values are coded with the darker colors. This non monotonicity of the relationship color/brightness creates ambiguities.

2.5. Selection of the Benchmark

In this section, we will define the benchmark to assess the performance of the algorithm. A standard reference to evaluate the predictions are the CODE predictions made by NOAA [23] (see Section 4.2). Another commonly used reference as benchmark predictor is either a prediction using the current frozen map or as a prediction the *cyclic* map, that is, the immediately preceding map of the same time as the time to be predicted. We will formally define the two predictors as follows:

- **Frozen**: $\hat{M}ap_{frozen}(t+\tau) = Map(t)$
- **Cyclic**: $\hat{M}ap_{cyclic}(t+\tau) = Map(t - 24\,\text{h} + \tau)$

In Section 4.2 we present the comparison with the NOAA forecast product.

As a benchmark in the following sections, we will use the cyclical prediction $\hat{M}ap_{cyclic}(t+\tau)$.

We argue this decision through Table 4, in which we show the prediction errors in RMSE (TECU) for prediction horizons ranging from 3 h to 48 h. In this case, one can see that the prediction cyclic $\hat{M}ap_{cyclic}(t+\tau)$ RMSE error and the standard deviation are constant regardless of the prediction horizon, and equal to the 24-h error of the frozen predictor $\hat{M}ap_{frozen}(t+\tau)$. This is to be expected since at all times the cyclic predictor behaves as a 24-h predictor. On the other hand, an important limitation of the use of the frozen prediction $\hat{M}ap_{frozen}(t+\tau)$ as a benchmark is that the comparison is made under non-comparable ionospheric conditions. This results in a sinusoidal behaviour of the RMSE, which increases from 3 h to 12 h and then decreases to a minimum at 24 h. This behaviour is then repeated, reaching a new minimum at 48 h. Therefore, since the frozen version $\hat{M}ap_{frozen}(t+\tau)$ is a very pessimistic benchmark, and has a component that depends on the time of day, we will use as a benchmark only the $\hat{M}ap_{cyclic}(t+\tau)$.

Table 4. Forecasting RMSE (TECU) for $\hat{M}ap_{frozen}(t+\tau)$ vs. $\hat{M}ap_{cyclic}(t+\tau)$ (June 2019).

Horizon: τ (hours)	**3 h**	**6 h**	**8 h**	**12 h**	**16 h**	**20 h**	**24 h**	**28 h**	**32 h**	**36 h**	**48 h**
$\hat{M}ap_{frozen}(t+\tau)$ (TECU)	1.87	2.35	2.51	2.59	2.51	2.18	1.42	2.19	2.57	2.61	1.54
$\hat{M}ap_{cyclic}(t+\tau)$ (TECU)	1.43	1.43	1.41	1.45	1.41	1.42	1.42	1.42	1.42	1.44	1.42

To get an idea of the differences between benchmarks and NNGIM prediction, in Figure 6 we present the comparison of the reference map (6-h a head ground truth), with the predictions using the NNGIM algorithms, the cyclic and the frozen reference. The cyclic reference provides local features of the TEC distribution similar to the reference map, while the frozen maps have a quite a different geographical distribution of TEC. On the other hand, the NNGIM prediction, despite using maps from other years, captures the structure of the TEC distribution of the reference map.

3. Results

For the analysis of the algorithm, we have selected two years of the *C24* cycle and two days of each year. The criterion for selecting the years was to have a sample of one year of high activity in the cycle and one year of low activity. Likewise for the days, in order to contrast the behaviour of the algorithm in the case of storm days vs. quiet days, we chose two storm days of each year and two adjacent days without a storm.

3.1. Analysis of Selected Years: 2015 and 2018

Figure 1 shows the time series of the average monthly TEC value for the two selected years. The first difference observed in the two years is the underlying monthly average TEC level and the fact that in the most active year (2015), the monthly profile of the TEC level has a marked cyclical component with a minimum in the summer. On the other hand, in the least active year (2018), the cyclical component has a lower amplitude. The mean annual TEC value for 2015 is 20 TECU, while in 2018 it is 8.8 TECU.

First, we show the performance of the NNGIM algorithm in TECU values and then for comparison purposes in percentages concerning the prediction using the frozen cyclic.

In Table 5 we show the average TECU prediction RMSE for 4 prediction horizons. In 2015 the prediction error increases as we increase the horizon from 17% to 20% of the average TEC value. On the other hand, the error in 2018 remains almost constant regardless of the horizon and stands at 18% of the average TEC value in that year. However, as we will see below, the prediction error has an annual cyclical component, being lower in the summer.

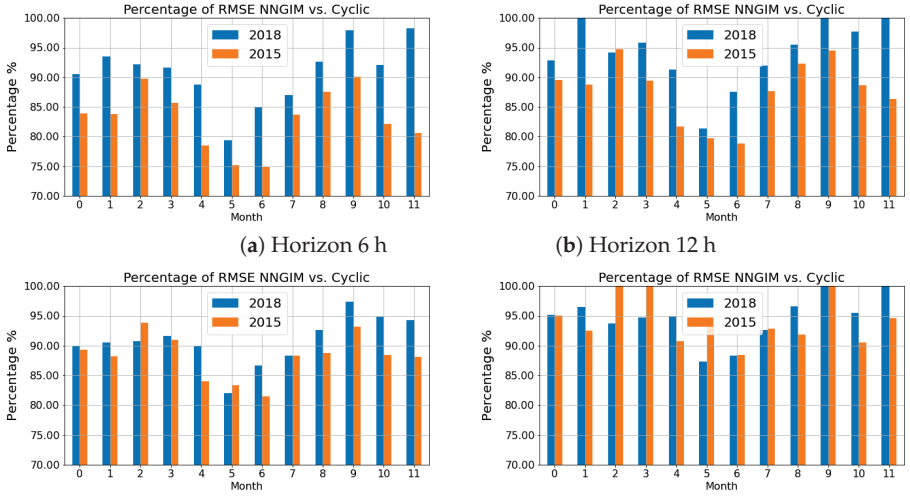
Table 5. RMSE error of the NNGIM algorithm for several horizons.

Horizon	6 h	12 h	24 h	48 h	Mean TECU
2015 (TECU)	3.50	3.70	3.72	4.00	**20.0** TECU
2018 (TECU)	1.59	1.66	1.59	1.66	**8.8** TECU

In Figure 7 we present the percentage change of the RMSE value for the cyclical prediction vs. NNGIM for various horizons. That is, we plot the ratio

$$\frac{\hat{M}ap_{nngim}(t+\tau)}{\hat{M}ap_{cyclic}(t+\tau)} \times 100\%$$

The first conclusion derived from the figures is that the use of NNGIM provides a decrease that follows an annual pattern and in the summer months for 6 and 12-h horizons provides a decrease in error in the order of 20% to 25%. This contrasts with the experience with tangent spaces predictions (see [3]) and deep learning based methods (see Section 1.2), where a significant degradation in quality is reported at prediction horizons of the order of 6 h. The prediction at 24 and 48 h reported as a percentage of frozen in [9] using deep learning is similar to the one shown in the lower row of Figure 7.



(**a**) Horizon 6 h (**b**) Horizon 12 h

(**c**) Horizon 24 h (**d**) Horizon 48 h

Figure 7. Percentage of RMSE reduction with regard to cyclic freezing for the horizons of 6 h, 12 h, 24 h, 48 h.

The 12-h forecast results are worse than the 24-h ones except for the months of May and June. This is because this is the moment in the interval $(t, t + 24\text{ h})$ when the ionosphere configuration is maximally different from the current state.

On the other hand, 48 h seems to be a natural limit for the method, as the error reduction for frozen cyclic is on an annual average of 95%.

3.2. Performance on Selected Days of 2015 and 2018

To evaluate the performance of the NNGIM method, we selected two days at the maximum of cycle 24 and two days at the minimum of the same cycle. The criterion for selecting the days was that one of them coincided with a geomagnetic storm and the other one coincided with a nearby day without significant activity. The selected days were:

1. 17 March 2015 (St.Patrick Day storm) and 5 March 2015 (non storm day).
2. 25–26 August 2018 (storm day) and 13–14 August 2018 (non storm day).

In both cases, the time distribution of geomagnetic activity index (i.e., Kp) are shown in Tables 1, 2, 6 and 7. The data was obtained from [24].

Table 6. Hourly Kp for the 5 March 2015.

Hour	**00–03 h**	**03–06 h**	**06–09 h**	**09–12 h**	**12–15 h**	**15–18 h**	**18–21 h**	**21–00 h**
Kp	1	0	0	1	2	2	2	1

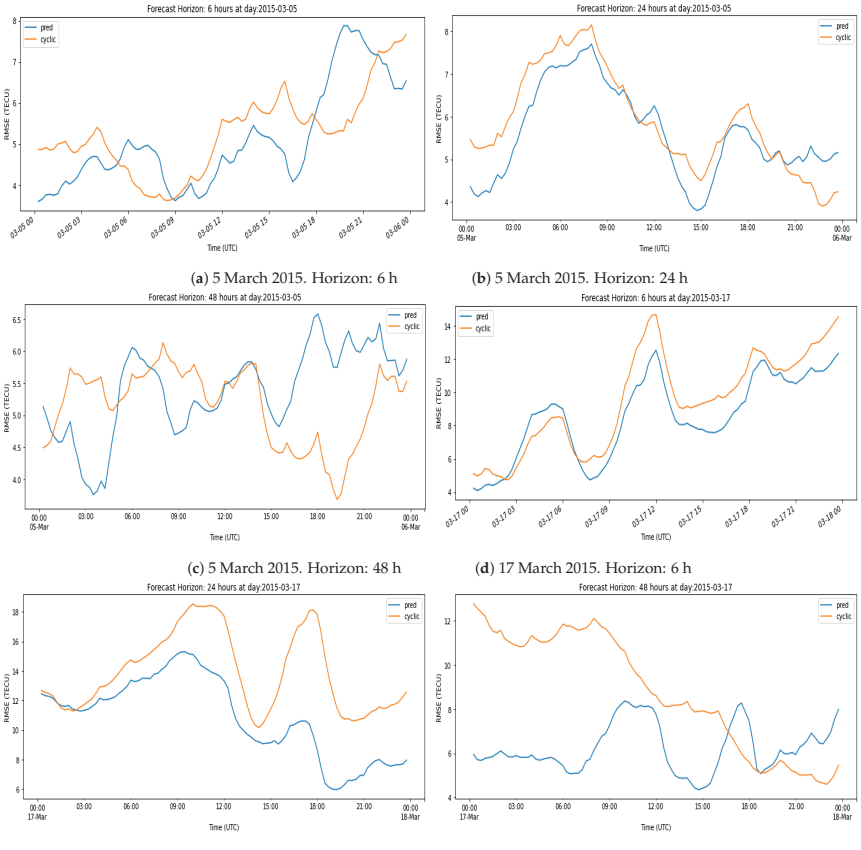
Table 7. Hourly Kp and 13,14 August 2018

Hour	**00–03 h**	**03–06 h**	**06–09 h**	**09–12 h**	**12–15 h**	**15–18 h**	**18–21 h**	**21–00 h**
Kp (13 August)	1	1	1	1	1	1	0	1
Kp (14 August)	2	1	1	1	1	0	0	2

3.2.1. Performance on 5 and 17 March 2015

In Figure 8 we present the comparison of the NNGIM predictor versus the cyclic frozen for various horizons in the form of a time series, at a rate of one map every 15 min.

In the top row, the performances of NNGIM vs. frozen cyclic are compared for the 5th of March 2015, which is a day with no significant events (see the Tables 1 and 6). The difference in performance is irregular for the 6-h forecast, while for the 24-h forecast the average reduction over the day is a little more than a 10% error. The worse behaviour towards the end of the day could be due to the increase of the Kp indicator and the presence of three solar flares in close temporal proximity. Since the NNGIM method assumes that similar situations have been seen in the past and are used for prediction, the changes in this particular configuration might not have been seen in the past.



(**a**) 5 March 2015. Horizon: 6 h (**b**) 5 March 2015. Horizon: 24 h

(**c**) 5 March 2015. Horizon: 48 h (**d**) 17 March 2015. Horizon: 6 h

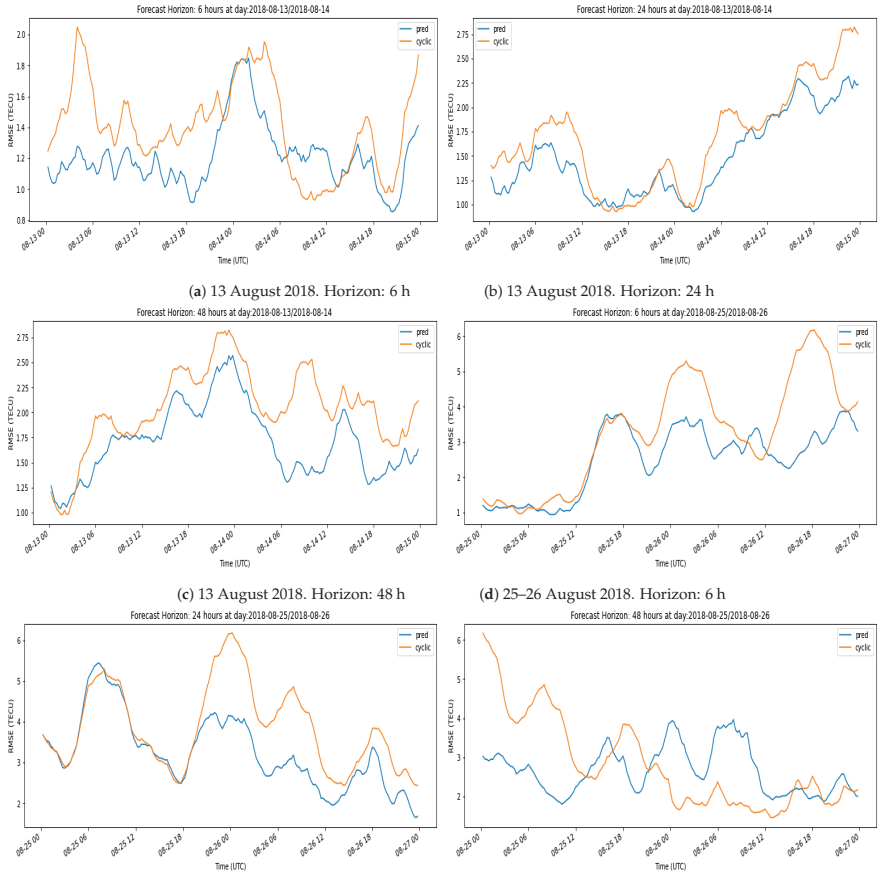
(**e**) 17 March 2015. Horizon: 24 h (**f**) 17 March 2015. Horizon: 48 h

Figure 8. Comparison of the NNGIM forecast vs. frozen cyclic RMSE. Upper row: 5 March 2015 (12 days before the storm). Lower row: 17 March 2015 (the St.Patrick storm day).

In the bottom row, we show the performance throughout the 17 March 2015 (Saint Patrick's Day storm). The RMSE level compared to the 5 March is between two and three times higher. However, in this case, the NNGIM predictor shows on average a better performance than the cyclic frozen with variations depending on the forecast horizon. For the first hours of the day, the NNGIM predictor performs similarly to cyclic frozen, for the 6 and 24-h horizons, improving throughout the day. An interesting behaviour is that at 48 h the RMSE of the NNGIM forecast remains at low levels throughout the day, while the frozen cyclic in the early hours provides twice the error.

3.2.2. Performance on 13–14 and 25–26 August 2018

Figure 9, shows the RMSE time series for the two selected days at a time of the low activity solar cycle. On that day, the RMSE level is similar to that of the 5th of March 2015 analysed above, which was a day of low geomagnetic activity, while being in a high activity phase of the solar cycle.



(**a**) 13 August 2018. Horizon: 6 h

(b) 13 August 2018. Horizon: 24 h

(**c**) 13 August 2018. Horizon: 48 h

(**d**) 25–26 August 2018. Horizon: 6 h

(**e**) 25–26 August 2018. Horizon: 24 h

(**f**) 25–26 August 2018. Horizon: 48 h

Figure 9. Comparison of the NNGIM forecast vs. frozen cyclic RMSE. Upper row: 13–14 August 2018 (12 days before the storm). Lower row: 25–26 August 2018 (storm day).

On 13–14 August 2018, the NNGIM prediction is better or equal to that of the cyclic frozen, except for a brief interval on the 14th of March at a 6-h horizon. The average improvement over the day is in the order of 25% for 6 h, 13% for 24 h, and 18% for 48 h. However, there are significant fluctuations throughout the day and the slopes/error patterns vary from horizon to horizon.

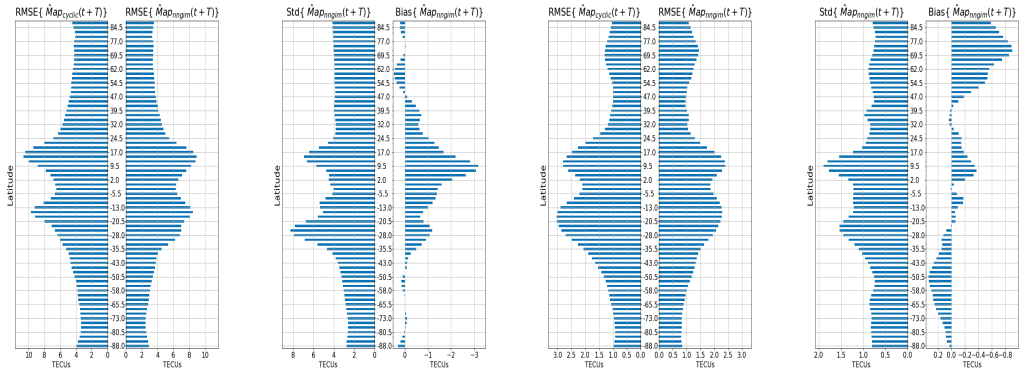
On 25–26 August 2018 (storm day) for the 6- and 24-h horizons NNGIM systematically performs better than the frozen cyclical. The performances at the 6- and 24-h horizons are practically the same for the 25th day, while they differ significantly for the 26th day, with NNGIM being 25–50% better over long time intervals.

3.3. RMSE, Bias and Standard Deviation by Latitude

In this section, we will study the relationship of RMSE with standard deviation and bias. In Figure 10, we show the performance for a horizon $T = 6$ h. In the Figure we present by latitude (a) the RMSE of the NNGIM and frozen cyclic predictions and (b) the standard deviation and bias components of the NNGIM. The study period consists of the dates studied above, i.e., August 2015 and May 2018. The values were calculated on 3007 maps corresponding to 31 days, with maps every 15 min.

The first observation is that the NNGIM prediction has a lower RMSE at all latitudes on the two studied dates. The RMSE maxima are located in the case of NNGIM at the

same latitude, while in the case of frozen cyclic the latitude in one case differs. On the other hand, the maxima in the standard deviation do not coincide with the RMSE maxima, noting that the difference is explained in the case of March 2015 by a very high bias at about 10 degrees north latitude. The bias of -3 TECU observed in this case is rare, in the maps observed by the author, the bias, in general, was less than 1 TECU, as illustrated in the case of August 2018.



(**a**) RMSE March 2015 (**b**) Std/Bias March 2015 (**c**) RMSE August 2018 (**d**) Std/Bias August 2018

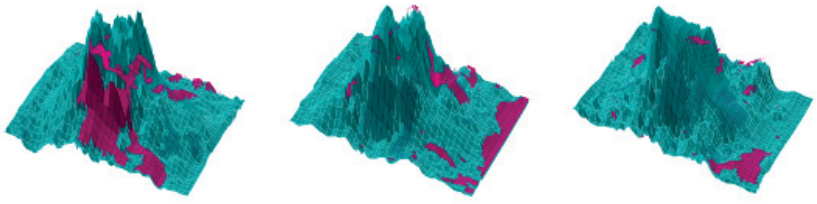
Figure 10. Performance for a horizon $T = 6$ h. RMSE, bias and standard deviation by latitude. (**a**) Comparison of the RMSE between the NNGIM and the frozen cyclic March 2015, (**b**) standard deviation and bias for the NNGIM March 2015, (**a**) Comparison of the RMSE between the NNGIM and the frozen cyclic August 2018, (**b**) standard deviation and bias for the NNGIM August 2018. Note that the bias and standard deviation are not the same scale.

4. Discussion

4.1. Reliability and Confidence Margins of the NNGIM Algorithm

In this section, we will study the reliability of the standard deviation estimated from the nearest neighbours provided by the algorithm. The purpose is to show that the standard deviation computed on the nearest future maps correctly represents the variability of the predicted map. We will show the reliability from two points of view, the first one consists of plotting several maps and showing the regions not covered by the confidence margin (defined as one standard deviation around the mean TEC value at each geographic coordinate of the GIM map, i.e., we associate a margin of about 68% confidence with each interval of 2.5 degrees latitude by 5 degrees longitude) given by the standard deviation provided by NNGIM. The second point of view will consist of showing the decrease in the error obtained when the prediction is considered to be included within the confidence margin given by the standard deviation.

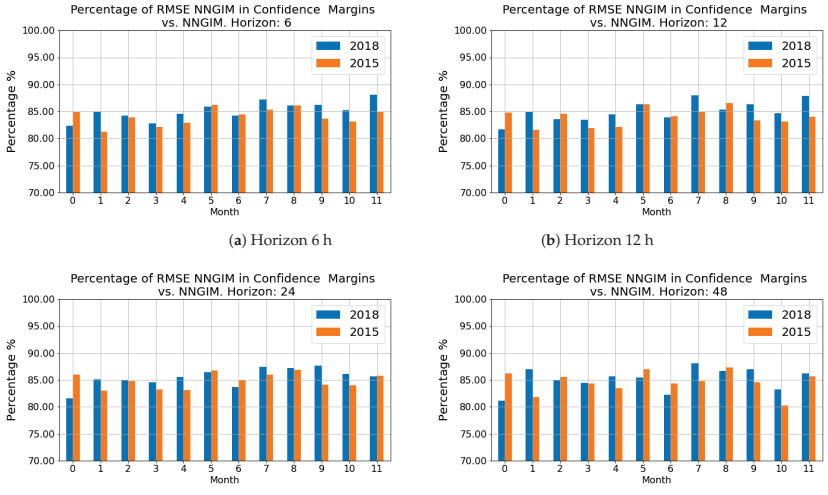
In Figure 11, we show maps for different dates for the month of June 2019, in which we mark in green the region covered by the interval $Forecast_{Map} \pm Forecast^{Std}_{Map}$, and in red the areas of the prediction that fall outside this interval. The images show that the areas of the $Forecast_{ref}$ maps not covered by a standard deviation margin are located in the periphery or at the areas of sharp transition.



(**a**) Map at 2019-06-29 23:00:00 (**b**) Map at 2019-06-22 17:45:00 (**c**) Map at 2019-06-12 10:15:00

Figure 11. Forecast maps in which the basemap coincides with the global coordinates (latitude ±90 degrees, longitude ±180), and the height shows measured TEC values. The colors distinguish, (green), the regions of the GIM map where the prediction is within the ± sigma range, and in red the regions where the prediction is outside. *Green areas*: show the areas where the reference $Forecast_{ref}$ is included in $Forecast_{Map} \pm Forecast^{Std}_{Map}$. *Red areas*: areas where $Forecast_{ref}$ is outside the margin.

In Figure 12 we show the error decrease regarding the NNGIM prediction if we consider only data outside the interval within the confidence margin. That is, we consider the error to be zero if the predicted map is contained in the margin, i.e., $Forecast_{ref} \subset Forecast_{Map} \pm Forecast^{Std}_{Map}$. It is seen that systematically for the two years and prediction horizons, the error decreases between 15 and 20%. In other words, assuming the correct value is within the confidence interval significantly reduces the error. An interesting feature is that this error reduction does not depend on either the season of the year or the prediction horizon.



(**a**) Horizon 6 h (**b**) Horizon 12 h

(**c**) Horizon 24 h (**d**) Horizon 48 h

Figure 12. Performance for $Forecast_{ref} \subset Forecast_{Map} \pm Forecast^{Std}_{Map}$. Percentage of RMSE reduction with regard to cyclic freezing for the horizons of 6 h, 12 h, 24 h, 48 h.

4.2. Validation of the Method with JASON3 and CODE Data

Next we show the results of the validation of the NNGIM VTEC in terms of the differences with respect to JASON3 VTEC measurements (see Figure 13) and the comparison with other GNSS VTEC products in terms of Bias, Variance and RMS (see Figure 14).

This part of the study was conducted in the interval of the first 100 days of the year 2021. Note that for the sake of completeness of the analysis of the method, we have performed the experiments at different times of the solar cycle. Given the space limitation, we think that in this way we can provide the maximum information of the algorithm from

the point of view of each issue to be evaluated. The CODE data was downloaded from the NOAA website [23].

The comparison was made between the products based on NNGIM prediction at 24 h (UN4G) and 48 h (UN8G), vs. IGSG and Center for Orbit Determination in Europe (CODE) VTEC prediction model products, at 24 h (C1PG) and 48 h (C2PG).

In Figure 13, we show the histogram of the VTEC residual defined as $\delta V = VTEC_{JASON3} - VTEC_{ForecastGIM}$ on a logarithmic scale to enhance the details in the low-density parts of the histogram, i.e., regions where the number of samples per bin is much lower than at the mode of the distribution. For comparison purposes on the figure, there is a summary of the relevant statistics of each product, i.e., bias, standard deviation, and RMS. Note that the Std. Dev and RMS of the NNGIM prediction at 24 h (UN4G) and 48 h (UN8G) are systematically lower than the CODE and IGSG. Note that the tails of the distributions are similar. Furthermore, the distribution related to the NNGIM product having a lower width compared with the CODE products. This indicates that the probability of a high-value positive error in the NNGIM products is much lower than the other products.

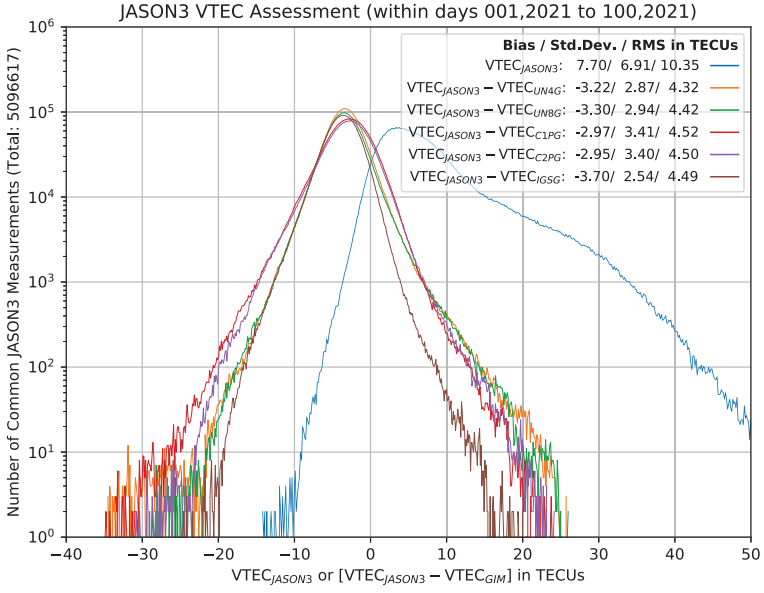


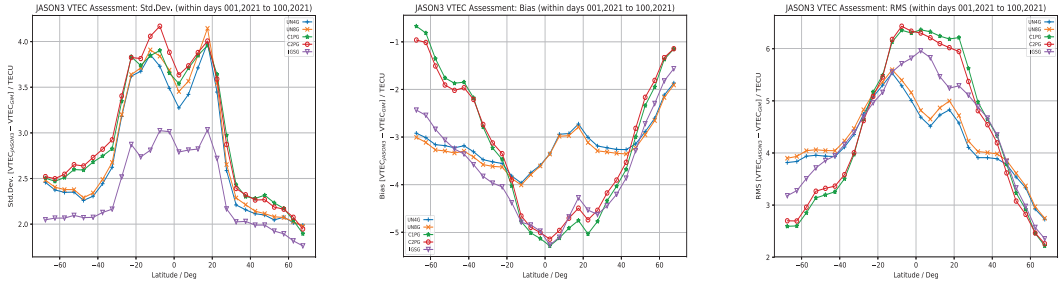
Figure 13. Histogram, in log scale for the number of counts, of VTEC difference of JASON3 measurement minus GIMs value for the first 100 days of 2021, the color code indicates the comparison for different forecasting products. The histogram of the reference values of JASON3 is represented in gray. The corresponding overall bias, standard deviation (Std.Dev.), and RMS are indicated in the upper right legend.

Next, we will compare, concerning the JASON3 measurements, the products by latitude, as a function of the differences in standard deviation, bias, and RMS.

In Figure 14, on the left, we show the standard deviation of the VTEC residual vs. JASON3 at 5-degrees longitudinal intervals. Note that the standard deviation is weighted by the number of JASON3 observations in cells in the same 5-degree latitude range. The 24-h prediction product based on NNGIM, UN4G consistently has a lower standard deviation than the equivalent CODE, C1PG except for the sample at 15 degrees latitude north where they are the same. The largest differences are observed at the equator and in areas of north/south latitude greater than 35 degrees. In the case of the 48-h forecast products (UN8G vs. C2PG), the trend is very similar, with NNGIM having a lower standard deviation at all latitudes except at 15 degrees north latitude.

In Figure 14, in the center, we show the bias of the products. In this case, the bias of the NNGIM products is lower, except in the region below −35 degrees south latitude and above 45 degrees north latitude. The explanation for this bias corresponds to the fact that there is a different ionosphere sampling model, as explained in [19].

Finally, in Figure 14, on the right, we show the RMS value by latitude, in this case, the RMS of the prediction is better for the NNGIM products between −30 degrees south latitude and 50 degrees north latitude. Note that from 50 degrees north latitude the difference concerning CODE is less than half a TECU, and on the other hand in the equatorial region the UN4G and UN8G products provide an improvement of 2 TECUs. The difference in the south polar region could be because there are fewer stations, and therefore the GIMs are less accurate.



(**a**) Standard deviation (**b**) Bias (**c**) RMS

Figure 14. Jason assessment for latitudinal zones, the color representing different products. Note that the measures are weighted by the number of JASON3 observations in cells with the same 5-degree intervals of latitude. Blue: UN4F, Orange: UN8G, Green: C1PG, Red: C2PG, Purple: IGSC.

Note that the availability of the NNGIM forecasting depends on the delay of generating the GIM maps, which is the case of the UPC-IonSAT is of about half an hour, while the availability of the CODE maps can be with a delay of up to 5 or 7 h, which makes the effective forecasting horizon shorter.

4.3. Considerations about the Quality Assessment by Means of JASON3 VTEC Measurements

The importance of the VTEC measures obtained by JASON3 lies in the fact that it provides us with an objective reference of the real value for the comparison purposes. The measures provided by JASON3 allow us to determine whether the estimate made by the prediction product provides a correct value or introduces biases. As the orbit altitude of JASON3 is about ∼1300 km, the altimeter can count almost all the VTEC of the ionospheric state above the ocean region. It is important to emphasize that over the ocean areas, the GIM used for the prediction might have large interpolating errors appearing due to their far distance from GNSS ground stations. Therefore, the use of JASON3 VTEC measurement allows for a critical evaluation of the forecast products in adverse circumstances. In this work, the raw observations of the JASON3 VTEC were preprocessed to reduce the measurement noise. The process carried out included the use of a temporal sliding window, removal of outliers, and so on, as explained in [25,26].

Evaluation using dSTEC may be an alternative for evaluating VTEC values of GIM prediction products. However, in this particular case the use of dSTEC may not be appropriate because of the following. Typically, the JASON3 VTEC assessment is a validation method for GIMs only over the ocean region, so it may be appropriate to consider the complementary assessment for GIMs over the land region, namely the dSTEC assessment, which compares the difference between the observed STEC along the phase-continuous satellite-station arc and the calculated STEC from GIM, see details in [25]. However, the usage of altimeter VTEC measurements to assess GIMs has been proven to be a good external

assessment procedure, consistent with other methods based on GNSS data (behaving similarly to the dSTEC test [25]) but independent from GNSS and globally distributed. These are the main reasons behind focusing on altimeter data, being the JASON3 the one available during the whole period of analysis, see the former studies that used JASON2, JASON1, and TOPEX altimeters.

4.4. Explanation of the Limitation of Saturating Nonlinearities

The learning algorithms used in LSTM type neural networks employ the gradient associated with internal nonlinearities of hyperbolic tangent or sigmoid type. Both nonlinearities, as illustrated (hyperbolic tangent case) in Figure 15 (right), saturate for large absolute values, and the derivative is zero. The consequence of this is that the gradient used for estimating the weights of the neural networks in the high value regions is practically zero, and therefore no learning takes place. In long tail distributions (e.g., Kp, Solar Flux, Magnetic Field Index proton density, EUV radiation, etc.), with a morphology as shown in Figure 15 left, and histograms with outliers. It is the case that the learning algorithms have precisely the null gradient in cases of greatest interest from the point of view of prediction. Therefore, the estimation of the neural network weights becomes zero in the cases of extreme activity.

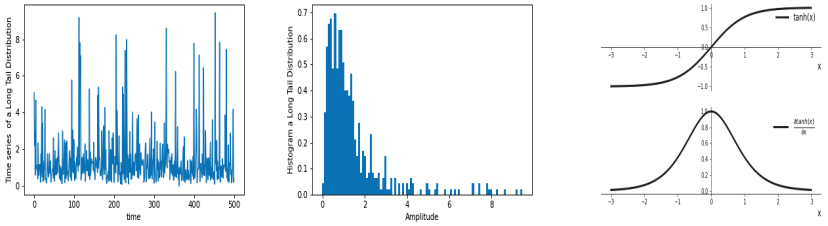


Figure 15. (**Left**) Example of a time series with a Long Tail distribution, (**Center**) Histogram of the time series, (**Right**) Comparison of the Tanh nonlinearity with its' derivative.

5. Conclusions

In this work, we have introduced a method to predict GIMs at various horizons based on the Nearest Neighbour technique. This technique allows predictors to be implemented without the need to train a model, and the computation time is small. The assumption on which the model is based is that a database covering more than one solar cycle is available, and that the geomagnetic conditions affecting the current map have somehow happened in the past, and that similar geomagnetic effects are distributed among several maps, whose linear combination allows a better approximation of the prediction. An advantage of the method is also that from the similar maps found in the historical database, a confidence margin can be created. The prediction using this confidence margin allows a significant decrease in the prediction error. We have performed a real-time implementation. The computational cost of adding a prediction horizon is very low, so in the implementation, predictions are made with almost no additional cost for arbitrary horizons. The prediction results improve compared to the frozen cyclic up to a 48-h horizon, which seems to be a natural barrier for this method. Finally, the method has been assessed in different moments of the solar cycle, taking into account days with storm and without significant geomagnetic perturbations. Additionally, the method has been assessed by comparing with the forecast at 24 and 48 h of the Center for Orbit Determination in Europe (CODE) prediction model products.

Author Contributions: Conceptualization, E.M.-M., H.Y. and M.H.-P.; methodology, E.M.-M., H.Y. and M.H.-P.; software, E.M.-M.; validation, E.M.-M., H.Y. and M.H.-P.; formal analysis, E.M.-M., H.Y. and M.H.-P.; investigation, E.M.-M., H.Y. and M.H.-P.; resources, M.H.-P.; writing—original draft preparation, E.M.-M., H.Y. and M.H.-P.; writing—review and editing, E.M.-M., H.Y. and M.H.-P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was been partially supported by the project PID2019-107579RB-I00 (MICINN) and done in the context of PITHIA NRF EU project. The work of Heng Yang was also partially supported by Natural Science Foundation of Chongqing, China (No. cstc2021jcyj-msxmX0191), by the Science and Technology Research Program of Chongqing Municipal Education Commission of China (Grant No. KJQN202101414), and by the Cooperative Projects between Undergraduate Universities in Chongqing and Institutes affiliated with Chinese Academy of Sciences (No.HZ2021014).

Data Availability Statement: The UQRG is openly accessible from IGS server (https://cddis.nasa.gov/archive/gnss/products/ionex/YEAR/DOY/uqrgDOY0.YYi.Z, accessed on 10 February 2022) and from UPC server (https://chapman.upc.es/tomion/rapid/YEAR/DOY_YYMMDD.15min/uqrgDOY0.YYi.Z, accessed on 10 February 2022) where YEAR and YY the four- and two-digit year identifiers, MM is month number, DD is day of month, and DOY is the day of year. Any missing file can be requested from the authors, in particular from Enric Monte Moreno (enric.monte@upc.edu).

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

References

1. García-Rigo, A.; Monte, E.; Hernández-Pajares, M.; Juan, J.M.; Sanz, J.; Aragón-Angel, A.; Salazar, D. Global prediction of the vertical total electron content of the ionosphere based on GPS data. *Radio Sci.* **2011**, *46*, 1–3. [CrossRef]
2. Web Page of IonSAT: IonSAT—Ionospheric Determination and Navigation Based on Satellite and Terrestrial Systems. Available online: https://futur.upc.edu/IonSAT?locale=en (accessed on 2 June 2021).
3. Monte Moreno, E.; García Rigo, A.; Hernández-Pajares, M.; Yang, H. TEC forecasting based on manifold trajectories. *Remote Sens.* **2018**, *10*, 988. [CrossRef]
4. Wang, C.; Xin, S.; Liu, X.; Shi, C.; Fan, L. Prediction of global ionospheric VTEC maps using an adaptive autoregressive model. *Earth Planets Space* **2018**, *70*, 1–14. [CrossRef]
5. Ian, G.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
6. Xiong, P.; Zhai, D.; Long, C.; Zhou, H.; Zhang, X.; Shen, X. Long Short-Term Memory Neural Network for Ionospheric Total Electron Content Forecasting Over China. *Space Weather* **2021**, *19*, e2020SW002706. [CrossRef]
7. Mustafa, U. Deep learning for ionospheric TEC forecasting at mid-latitude stations in Turkey. *Acta Geophys.* **2021**, *60*, 589–606.
8. Wen, Z.; Li, S.; Li, L.; Wu, B.; Fu, J. Ionospheric TEC prediction using Long Short-Term Memory deep learning network. *Astrophys. Space Sci.* **2021**, *366*, 3. [CrossRef]
9. Cherrier, N.; Castaings, T.; Boulch, A. Forecasting ionospheric Total Electron Content maps with deep neural networks. In *Proceedings Conference Big Data Space (BIDS)*; ESA Workshop: Toulouse, France 2017 .
10. Liu, L.; Zou, S.; Yao, Y.; Wang, Z. Forecasting global ionospheric TEC using deep learning approach. *Space Weather* **2020**, *18*, e2020SW002501. [CrossRef]
11. Ji, E.Y.; Moon, Y.J.; Park, E. Improvement of IRI global TEC maps by deep learning based on conditional Generative Adversarial Networks. *Space Weather* **2020**, *8*, e2019SW002411. [CrossRef]
12. Krankowski, A.; Kosek, W.; Baran, L.W.; Popinski, W. Wavelet analysis and forecasting of VTEC obtained with GPS observations over European latitudes. *J. Atmos. Sol.-Terr. Phys.* **2005**, *67*, 1147–1156. [CrossRef]
13. Weber, G.; Dettmering, D.; Gebhard, H. Networked transport of rtcm via internet protocol (ntrip). In *A Window on the Future of Geodesy*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 60–64.
14. Hernández-Pajares, M. Inputs received from Ionospheric research groups on activities related with RT global electron content determination. In Proceedings of the IGS RT WG Splinter Meeting, Pasadena, CA, USA, 23 June 2014; pp. 1–15.
15. Hernández-Pajares, M.; Juan, J.; Sanz, J. New approaches in global ionospheric determination using ground GPS data. *J. Atmos. Sol. Terr. Phys.* **1999**, *61*, 1237–1247. [CrossRef]
16. Orús, R.; Hernández-Pajares, M.; Juan, J.; Sanz, J. Improvement of global ionospheric VTEC maps by using kriging interpolation technique. *J. Atmos. Sol. Terr. Phys.* **2005**, *67*, 1598–1609. [CrossRef]
17. Roma, Dollase, D.; López Cama, J.M.; Hernández, Pajares, M.; García Rigo, A. Real-time Global Ionospheric modelling from GNSS data with RT-TOMION model. In Proceedings of the 5th International Colloquium Scientific and Fundamental Aspects of the Galileo Programme, Braunschweig, Germany, 27–29 October 2015.
18. Hernández-Pajares, M.; Juan, J.; Sanz, J.; Colombo, O.L. Application of ionospheric tomography to real-time GPS carrier-phase ambiguities resolution, at scales of 400–1000 km and with high geomagnetic activity. *Geophys. Res. Lett.* **2000**, 27, 2009–2012. [CrossRef]
19. Yang, H.; Monte-Moreno, E.; Hernández-Pajares, M.; Roma-Dollase, D. Real-time interpolation of global ionospheric maps by means of sparse representation. *J. Geod.* **2021**, *95*, 1–20.
20. Wang, L.; Zhang, Y.; Feng, J. On the Euclidean distance of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1334–1339. [CrossRef] [PubMed]
21. Omohundro, S.M. *Five Balltree Construction Algorithms*; International Computer Science Institute Technical Report; International Computer Science Institute: Berkeley, CA, USA, 1989.

22. NNGIM Forecasts at Different Horizons. Available online: http://chapman.upc.es/.trial_urtg/tomion/forecast_nngim/quick/last_results/ (accessed on 2 June 2021).
23. Code Data. Available online: ftp://ftp.nodc.noaa.gov/pub/data.nodc/ (accessed on 2 June 2021).
24. Space Weather Live. Available online: https://www.spaceweatherlive.com/ (accessed on 2 June 2021).
25. Hernández-Pajares, M.; Roma-Dollase, D.; Krankowski, A.; García-Rigo, A.; Orús-Pérez, R. Methodology and consistency of slant and vertical assessments for ionospheric electron content models. *J. Geod.* **2017**, *91*, 1405–1414. [CrossRef]
26. Roma-Dollase, D.; Hernández-Pajares, M.; Krankowski, A.; Kotulak, K.; Ghoddousi-Fard, R.; Yuan, Y.; Li, Z.; Zhang, H.; Shi, C.; Wang, C.; et al. Consistency of seven different GNSS global ionospheric mapping techniques during one solar cycle. *J. Geod.* **2018**, *92*, 691–706. [CrossRef]

electronics

MDPI

Article

IoT Monitoring and Prediction Modeling of Honeybee Activity with Alarm

Nebojša Andrijević [1,*], Vlada Urošević [1], Branko Arsić [2], Dejana Herceg [3] and Branko Savić [4]

[1] Faculty of Technical Sciences Čačak, University of Kragujevac, Svetog Save 65, 32000 Čačak, Serbia; vlade.urosevic@ftn.kg.ac.rs
[2] Faculty of Science, University of Kragujevac, Radoja Domanovića 12, 34000 Kragujevac, Serbia; branko.arsic@pmf.kg.ac.rs
[3] Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia; vuletic@uns.ac.rs
[4] Higher Education Technical School of Professional Studies, Školska 1, 21000 Novi Sad, Serbia; savic@vtsns.edu.rs
* Correspondence: andrijevicnebojsa@gmail.com

Abstract: A significant number of recent scientific papers have raised awareness of changes in the biological world of bees, problems with their extinction, and, as a consequence, their impact on humans and the environment. This work relies on precision beekeeping in apiculture and raises the scale of measurement and prediction results using the system we developed, which was designed to cover beehive ecosystem. It is equipped with an IoT modular base station that collects a wide range of parameters from sensors on the hive and a bee counter at the hive entrance. Data are sent to the cloud for storage, analysis, and alarm generation. A time-series forecasting model capable of estimating the volume of bee exits and entrances per hour, which simulates dependence between environmental conditions and bee activity, was devised. The applied mathematical models based on recurrent neural networks exhibited high accuracy. A web application for monitoring and prediction displays parameters, measured values, and predictive and analytical alarms in real time. The predictive component utilizes artificial intelligence by applying advanced analytical methods to find correlation between sensor data and the behavioral patterns of bees, and to raise alarms should it detect deviations. The analytical component raises an alarm when it detects measured values that lie outside of the predetermined safety limits. Comparisons of the experimental data with the model showed that our model represents the observed processes well.

Keywords: IoT monitoring; predictive modeling; honeybees activity; precision beekeeping

Citation: Andrijević, N.; Urošević, V.; Arsić, B.; Herceg, D.; Savić, B. IoT Monitoring and Prediction Modeling of Honeybee Activity with Alarm. *Electronics* **2022**, *11*, 783. https://doi.org/10.3390/electronics11050783

Academic Editors: Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Received: 3 February 2022
Accepted: 19 February 2022
Published: 3 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Although humanity is constantly advancing technologically, this development influences the environment, inevitably changing it both intentionally and unintentionally. Nature runs its course, and our influence disturbs the normal natural processes, changing the balance of natural perfection. Modern approaches in agriculture, the application of pesticides, herbicides, other chemical agents, and artificial pollinators, the flowering of nature in periods, and untimely conditions have changed the ecosystem of nature itself and bee societies, which is the topic of our research. In addition to the aforementioned, diseases of bee colonies, Varroa destructor infections, the effects of pesticides and herbicides, lack of food in hives, the loss of the queen and significant losses caused by unusual changes in the environment, meteorological conditions, and the winter season also contribute to beekeeping problems. The challenge was to design and manufacture an improved monitoring and data analysis system that would process data with advanced data analysis techniques on the basis of experience gained from previous studies.

We cannot influence events in the environment of bees occurring in nature, but we can monitor, measure. and collect data. With the methodological application of software

algorithms, we can create prediction systems and learn their patterns. Artificial events and problems can be monitored, measured, and recorded, and we can influence them in order to control the outcome and prevent damage caused by human influence.

Such problems have been studied for years, and methodological approaches have changed. The development of precision microelectronics and the application of advanced methodological approaches have enabled a new way of approaching data measurement and analysis. The use of such systems enables mapping the causes of problems, and creating approaches to prevent and solve them. In this way, we come closer to solving the problems of modern apiculture.

In order to solve these problems, and relying on previous research, we created a complex system that thoroughly monitors and measures an expanded set of parameters that were proven through the mentioned research as relevant factors of influence. The created system opens possibilities for multiple and different approaches to processing and analyzing obtained results. An activity of bees (exits and entrances) represents a sequence of data points that occur in successive order over a period of time, which gives us the ability to apply algorithms for time-series forecasting. Time-series forecasting is the process of analyzing time-series data using statistics and modeling to predict and inform strategic decision making. The recent frequent use of recurrent neural networks is noticeable because they show high performance in achieving this task. Taking into consideration the data with which we work, approaches from the state of the art to more advanced and complex approaches were tested. The best results for bee activity forecasting are achieved by using recurrent neural networks (LSTM cell).

Precisely collected and processed data from such a system give insight to the beekeeper regarding the situation inside and around the hive. Obtained results by analyzing and applying the prediction model trigger alarms and inform beekeepers about a change in circumstances in the hive, suggesting the application of adequate solutions to prevent potential (predictive) problems. Thus, they can act in a timely and preventive manner on hives or bees in order to avoid a negative outcome.

Software solutions in the form of an application based on artificial intelligence must be accompanied by hardware components, which is the key to the presented management system and this work. Through the research, we realized that hive management is the next level in precision beekeeping, and the implementation of IoT MAP systems for supervising and controlling hives is inevitable for the architecture of the solution.

A comprehensive, all-inclusive discussion of the considered topic, as presented in this paper, consists of a set of parts, each one with an individual contribution. However, the system is presented as a whole with the following summary of contributions:

- a system for bee movement monitoring was constructed and installed on the basis of which we could correlate independent and dependent indicators;
- a large set of sensors for monitoring conditions from within and outside the hive was installed, which collects a wide array of real-time parameters;
- a microcontroller-based IoT device was designed and constructed, which aggregates sensor readings and uploads data to the cloud;
- an AI-based computational module was created and deployed to the cloud backend, which enables real-time analytical and predictive assessment of data uploaded from the IoT device;
- a web frontend app was designed and created, which enables insight into real-time data from sensors at the hive and results from the AI module, namely, analytical and predictive warnings and alarms.

All listed components work as an integrated system that gives beekeepers and biologists insight into the wellbeing of bees, and allows for the monitoring of their behavioral patterns. Data are observed and analyzed depending on meteorological conditions, time of day, season, etc. Future work includes the possibility for taking actions such as hive entrance shutdown, ventilation, suggestions for hive relocation, and engagement of the automatic feeder.

All components of the proposed system are described in detail here. Section 2 lists relevant references on the topic of beekeeping that discuss various parameters impacting the behavior and wellbeing of bees. Section 3 describes the hardware of the constructed IoT microcontroller station and connected sensors. Section 4 contains the description of the web application for the real-time data monitoring and display of warnings. Section 5 describes the relevant data that were collected from the hive for the construction of the AI model. Section 6 contains the description of the applied predictive models. Section 7 contains the experimental results, and Section 8 is the conclusion.

2. Related Work

In recent years, beekeepers have encountered problems with mass bee deaths [1,2] and bee migrations due to climate change, and the impact of weather conditions on flowering disorders in nature in periods when bees should collect pollen. The use of various pesticides and herbicides in plant protection, and spraying at a time when bees are active in periods of disturbed climatic conditions due to high temperatures and humidity cause bees to be active in the later hours when spraying is performed [3]. This study indicates an advanced solution that could be applied for the intelligent monitoring of events around the bee habitat. It encompasses constant monitoring inside and outside a hive, real-time application, and artificial intelligence that includes a large number of dependent and independent factors influencing bee's life in the analysis. Previous works [4–8] provide an excellent introduction to the issue and indicate a wide range of approaches, proposals, and analyses of various data, and the necessity and importance of including the influence of many factors on the movement and life of bees. The aim of most works was to fully understand the movement, work, and life of bees living in apiculture (the hive), and which beekeeper takes care of the bees, so that they can raise, nurture, and monitor them with constant insight into the condition of bees in the hive. In that way, the beekeeper could quickly react to changes through alarms that would be triggered from intelligent monitoring if situations occur with sudden changes, deviations, or potential problems predicted by the solution from this paper.

Precision beekeeping [9–14] is a term that has appeared in recent years referring to the development of online tools for the continuous monitoring and control of bee behavior using an individual approach to society, avoiding exposing bees to additional stress and unproductive activities. As monitoring each bee colony requires expensive resources and is complex, precision beekeeping offers a solution in the form of monitoring individual bee colonies and their immediate environment.

The mentioned works include important factors indicating their individual value, such as temperature and humidity, and their influence on swarming or feeding [15,16], ref. [17–19] vibration and sound [20–26], the presence of gases [27,28], rain and wind [29], the amount and intensity of daylight, and UV and IR radiation indices; this paper covers all these factors together. There are also time series of recording and data collection, which were performed in hourly or daily time series in the mentioned works. The choice of hardware solutions that affect the accuracy of data in previous analyses and approaches [15, 27,30,31] differs from the approach in this paper, where we relied on advanced methods and data analysis. It is very important that analysis includes all dependent and independent influencing factors due to the complexity of the obtained results and different methods of inference.

The significance of constant monitoring [32–38] in terms of monitoring and measuring various parameters of influence within and immediately around the hive is shown in a large number of papers [39–48]. Most papers relied on the application of IoT technologies in designing these systems [19,49–58].

Regarding the application of artificial intelligence, the authors in [7] used a decision tree algorithm to classify the state of the hive. In order to maximize the identification of crucial colony activities, including healthy and unhealthy conditions, ten hive status classes were selected for this multi-class classification task. Our AI approach differs from the

mentioned one, because we try to solve the regression problem and to draw a conclusion about the status of the hive based on the activity of the bees, i.e., whether the conditions in the hive are healthy or unhealthy. Similar approach was presented in [59,60] where the deep neural networks were used to classify bee swarm activity from audio signals.

The monitored parameters are on a broad spectrum to indicate even the slightest significance of any time element, or any deviation or disturbance in relation to the natural environment in which bees normally function.

Parameters inside and outside the hive were monitored in very precise sequences of temperature, humidity, air quality (presence of various gases, smoke, carbon monoxide, etc.), noise, presence of different frequencies of sounds, shocks, vibrations, UV factors, IR factors, intensity and variations of daylight, wind intensity, all in correlation with the frequency (entrance and exits) of bees. Furthermore, one of the goals of the research was to use this system to indicate the range of influences of different factors and parameters, their intensity, and the mutual correlation of factors.

3. System Overview

The system consists of several hardware and software components (Figure 1). The main IoT unit, located at the hive, collects data from multiple sensors in and around the hive, and from a bee counting circuit located at the hive entrance. The main unit is based on Arduino Mega 256 and ESP32 microcontroller boards. Data from the sensors and the bee counting circuit are timestamped and transmitted to the cloud database via a cellular modem. In order to prevent data loss, they are also saved on a local memory card.

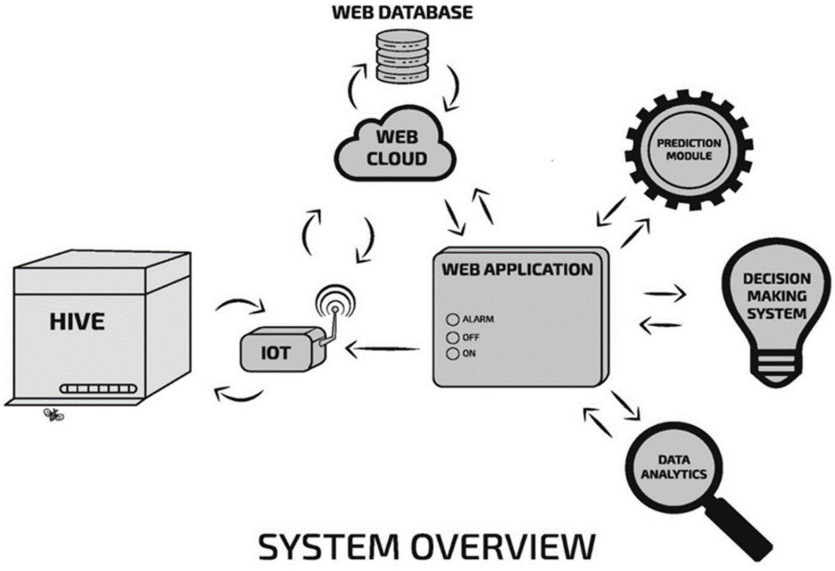


Figure 1. System overview.

A web application connects to the cloud database to enable the display of real-time and historical data. A decision-making system (DMS) also runs on the server, performing real-time data analysis and parameter prediction. This component can detect deviations from nominal parameters and accordingly generate alarms.

There are numerous solutions and tools for monitoring the movement of bees inside and outside the hive based on semiconductors, optical sensors, and photoresistors, for example [61,62], Arnia [63], Beecheck [64], the bee counter [65], and the honeybee counter [66]. Some solutions have exhibited problems due to a chosen approach to counting. The bee

counting circuit presented in this work is based on a set of two photoreflecting resistors per gate, where both resistors must be triggered to detect one pass. Depending on the order of activation, the direction of movement in or out of the hive is determined. In order to avoid congestion, the circuit contains 24 gates, enabling bees to simultaneously enter and exit through all of them.

The precise measurements of bee movements are the basis for reaching conclusions about the condition of hives, and they are related to every action. However, to obtain more precise movement results, bee movement data must be tied to dependent and independent variables inside and outside the hive.

An active beehive with the IoT main unit and sensors used for data collection in this research is shown in Figure 2.

Figure 2. Experimental hive setup with sensor electronics.

Main Unit Architecture

As seen in Figure 3, the scheme of the data collection system consisted of a microprocessor -controlled IoT base station.

The base station was connected to sensors for measuring parameters and data. The sensor sets in charge of the conditions in the hive were specially arranged in several levels following the structure of the frames and floor in the hive. Sensors for measuring parameters outside the hive were placed in the outer part of the system, but they are protected from direct meteorological influences that could lead to measurement errors. The bee counting sensor array is located at the entrance to the hive, where there are gates with photoresistors for the passage of bees to detect their movement. The entire system is controlled by the main unit microprocessor, which communicates with microcontrollers and initiates the collection of data that are forwarded via GPRS to the cloud system and web database. The data are also written into local storage.

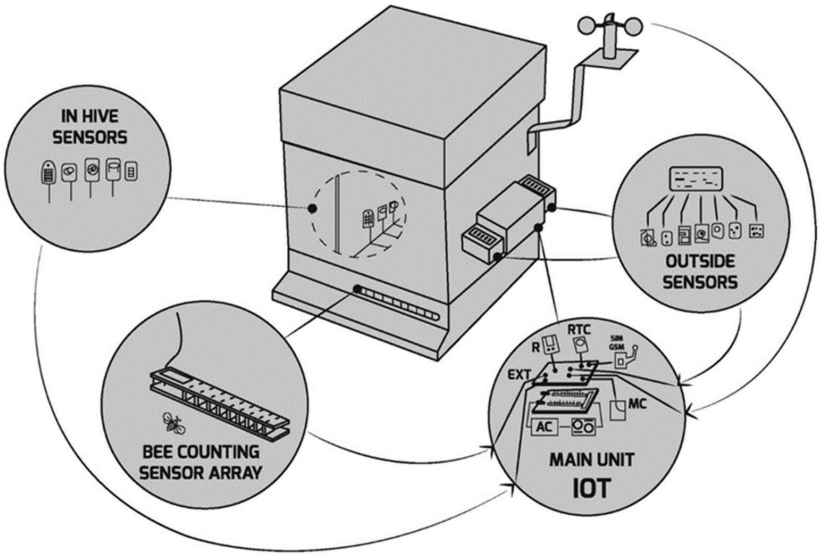


Figure 3. Schematic of data collection system.

The system consists of sensors and microelectronic components, preferably designed to avoid interruptions in operation, since it involves a large number of sensors and auxiliary modules operating at different voltage levels. The system was designed with low power consumption in mind, enabling a self-sustainable operation via solar power and a battery.

Figure 4 shows the schematic of the hardware and the main unit of IoT base station, where the central component is an Arduino Mega microcontroller, expanded with an extension module to accommodate all necessary electrical connections. Most sensors were attached over the industry-standard SPI and I2C buses available on the Arduino Mega.

The system was installed in such a way to avoid disturbing the bee ecosystem and prevent the impact of direct exposure to the weather in order to avoid measurement errors. For example, individual sensors that are exposed to direct sunlight are protected by clear glass without UV stabilizers to avoid measurement errors. Sensors for the detection of gases, frequencies, and noise were placed in such a way that they could record without interference and without being affected by weather conditions or direct sunlight.

The bee counting sensor array consists of devices for detecting the frequency of the movement of bees at the entrance to the hive in several corridors in order to smoothly monitor the movement of entering and leaving the hive. These are photoreflective resistors in which reflection is interrupted during movement; thus, the direction of movement is detected. The ESP32 microcontroller board controls the operation of these sensors.

Sensors inside the hive were positioned in such a way that they could function without the danger of being obstructed by bee wax, as bees wax any unknown elements inside the hive to protect the colony.

Collected data from the hive represent the microclimate of the living environment of bees and are valuable because they allow for the differences in measurements with values obtained outside the hive to be observed.

In the background of the main unit of the IoT system that collects data from the bee counting array and the measurement system with sensors there is a trained algorithm in charge of eliminating errors in measurements if they occur.

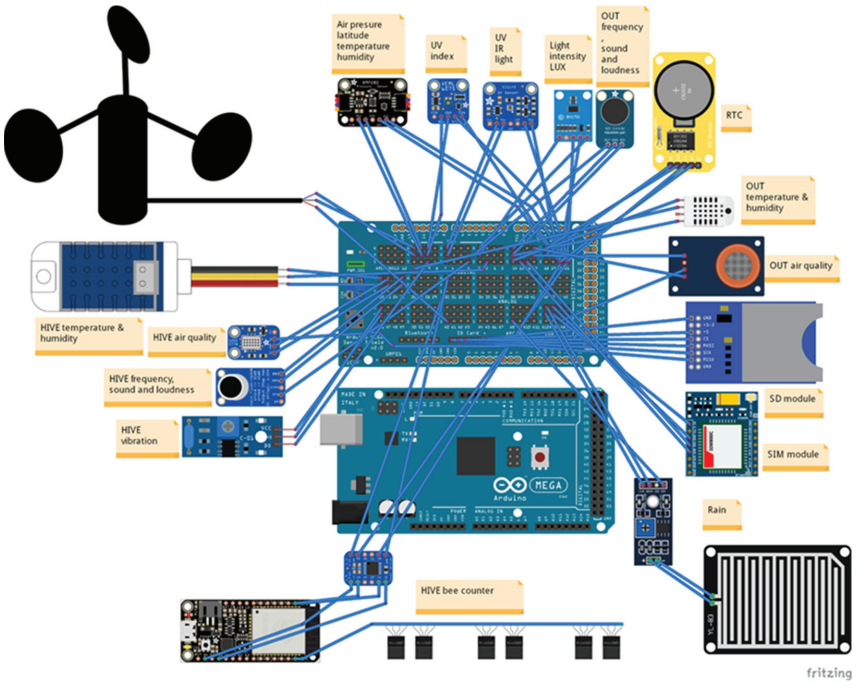


Figure 4. System schematic (created in Fritzing [67]).

4. WebAPP for MAP

A web application was developed that shows current sensor readings from the hive (Figure 5). Measured values are stored in a web database in real time.

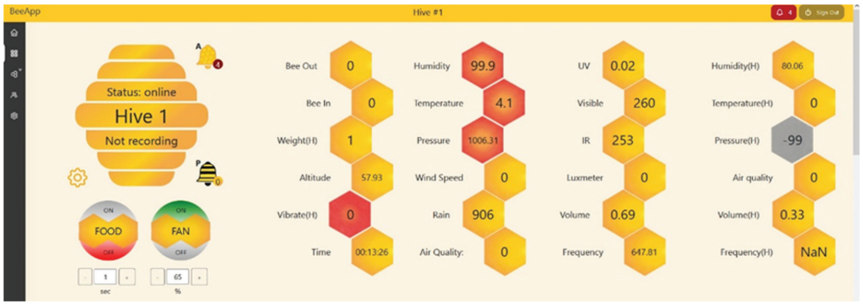


Figure 5. Application interface.

The web interface contains indicators for predictive and analytical alarms. The predictive alarm is activated by the prediction algorithm on the basis of bee movement (more details in the following section).

The movement of bees is most often caused by feeding, meteorological factors, daily activities in relation to the same factors, and human activity. In this way, we formulated directly and indirectly dependent factors, and their interdependence.

Cells that display values in the application are dynamic and change colors in relation to the displayed values. The analytical alarm is triggered when the measurement value approaches the critical value, for example, in the case of high temperature and high humidity. When the temperature value inside the hive exceeds 35 °C [68] or the relative

humidity is nearly 90%, the alarm is triggered. A push notification is sent informing about changes in the hive.

Before the predictive modeling (AI) module is described, an overview of the collected dataset from previous steps is provided. All collected variables, and which sensor is responsible for collecting which data are described. Additionally, required data cleaning and variable transformations are described.

5. Dataset Description

Data were collected during fall months, but in the paper, 20 successive days in October 2021 were used for analysis. Measurements were performed in 5 min intervals. Taking into consideration a small number of exits from the hive and small changes in weather conditions in a period of 5 min, especially in the observed period, time intervals were consolidated into 24 h. Final input features were obtained as the average value of all related values that belonged to the observed hour. Output values were obtained as the sum of all exits and entrances to the hive in that hour. Table 1 shows the structure of the dataset with all used variables and their descriptions. The entire dataset and code source is publicly available at https://gitlab.com/mali_banekg/beeactivityforecast (accessed on 15 January 2022).

Table 1. Sensors, control sensors, and description of measurements.

Column Name	Description	Column Name	Description
Date	Date of measuring	BME280_humi	Outside humidity control
Hour	Time of measuring	BME280_pressure	Air pressure
AM2302_1_Temp	Outside air temperature	BME280_alt	Altitude
AM2302_1_Humi	Outside humidity	SI1145_visible	Daylight intensity
AM2302_2_Temp HIVE	Temperature in a hive	SI1145_IR	Infrared intensity
AM2302_2_Humi HIVE	Humidity in a hive	SI1145_UV	UV index control
SW420_Vibrate HIVE	Vibrations in a hive	ANEM_voltage	Wind force
MHRD_rain	Rain sensor	ANEM_windSpeed	Wind speed
MQ135_PPM	Air quality sensor	MIC1_freq	Frequency spectrum
MICS6814_PPM	Air quality sensor	MIC1_volume	Sound level and loudness
MICS5524_PPM HIVE	Air quality sensor in a hive	MIC2_freq HIVE	Frequency spectrum in a HIVE
BH1750_lux	Day light and lux intensity	MIC2_volume HIVE	Sound level and loudness in a HIVE
VEML6750_uvindex	UV index	BEECNT_message OUT	Bee counter OUT HIVE
BME280_temp	Ouside temperature control	BEECNT_message IN	Bee counter IN HIVE
Date	Date of measuring	BME280_humi	Outside humidity control
Hour	Time of measuring	BME280_pressure	Air pressure
AM2302_1_Temp	Outside air temperature	BME280_alt	Altitude
AM2302_1_Humi	Outside humidity	SI1145_visible	Daylight intensity
AM2302_2_Temp HIVE	Temperature in a hive	SI1145_IR	Infrared intensity

The *BEECNT_message OUT* variable represents the number of bees that came out of the hive, while *BEECNT_message IN* represents the number of bees that entered the hive. These two variables were used as output in our models. In this way, we connected dependent and independent indicators of bee movement.

Counting the bees' entrances to and exits from the hive, and measuring the environmental conditions inside and outside of the hive are important for alarm initialization, and complement the results of other parameters that indicate the frequencies of movement of

bees obtained from the sensory measurements of the immediate environment. Without measuring all the above factors, especially weather conditions, the number of bees in and out would not be of greater significance and would only be a statistical detail. All these variables collected from the environment to which the bees belong could be used for a model development that can very precisely predict bee movements.

Some of the used variables and output variable Bee_IN (y) are shown in Figure 6. On the basis of the time-series shape, it is obvious that some of these variables were important for our model, such as lux or outside humidity (when the humidity value is high, the bees do not leave the hive).

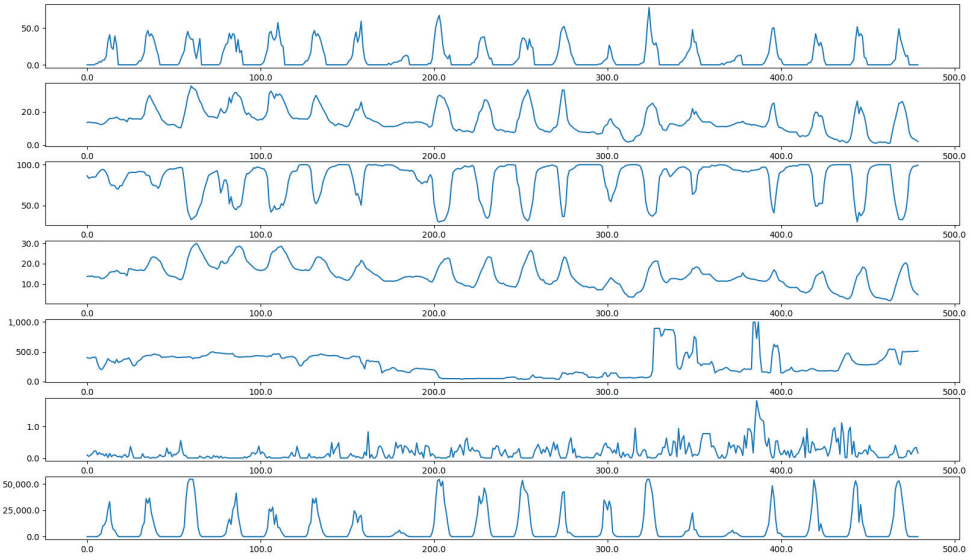


Figure 6. Shape of seven time series corresponding to output and inputs.

During the feature engineering phase, we took the advantage of the fact that there are periods during the day (24 h) when bees are not active, and a new binary variable is created that represents part of the day, daylight (5 a.m. to 6 p.m.) or night (other hours). If any activity was detected during the night period, we replaced them with the 0 value. This could usually happen around the border hours, when few exits or entries are detected.

Output columns BEE_IN and BEE_OUT were transformed by using the square root transformation (this is the so called power transformation) because in time-series analysis, this transformation is often considered to stabilize the variance of a series. Logarithmic transformation was skipped because some of the values were equal to 0. Figure 7 shows the original time series of the bee exits (red) and the time series after square root transformation is applied (blue). It is obvious that the number of outings on certain days was drastically reduced. This can be explained by the fact that weather conditions were probably worse that day, for example, it was raining or it was windy.

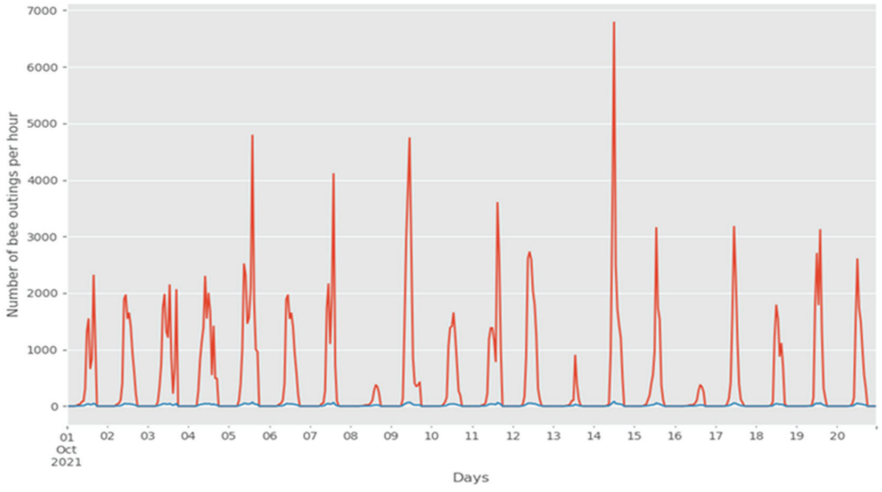


Figure 7. Number of bees counted per hour for 20 days.

6. Methodology

Time series data is a collection of observations obtained through repeated measurements over time, as is the case here. Unlike regression predictive modeling, time series also adds the complexity of a sequence dependence among the input variables. The recurrent neural networks are a powerful type of neural network designed to handle sequence dependence. The principal advantage of RNN over ANN is that RNN can model a collection of records (i.e. time collection) so that each pattern can be assumed to be dependent on previous ones. On the other hand, comparisons against ETS (error, trend, seasonal) and ARIMA demonstrate that (semi-) automatic RNN models are not silver bullets, but they are nevertheless competitive alternatives in many situations [69].

In this paper, we tested above-mentioned approaches, which are the most common and the most promising methods in time-series forecasting, in order to predict bee exits from and entries to the hive. This information may be important during periods when fruits and vegetables are sprayed, so that we can close bee hives when high activity is expected. First, we started from traditional approach ARIMA [70]. ARIMA is an acronym that stands for autoregressive integrated moving average, which is a generalization of the simpler autoregressive moving average that adds the notion of integration. After that, we tested two more advanced approaches, Facebook Prophet [71] and recurrent neural networks (LSTM) [72]. In the following subsections, a short description of these techniques is given. For more details, we refer readers to the original papers.

6.1. ARIMA

An ARIMA model is a class of statistical models for analyzing and forecasting time-series data. The model is fitted to time-series data to either better understand the data or predict future points in the series, known as forecasting. The model acronym was obtained after the key aspects of the model itself:

- AR: Autoregression. A model that uses the dependent relationship between an observation and some number of lagged observations.
- I: Integrated. The use of the differencing of raw observations (i.e., subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
- MA: Moving average. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

A standard notation used for ARIMA is ARIMA (p,d,q), where parameters p, d, and q can only be integer numbers denoting the lag order (number of lag observations included in the model), the degree of differencing (number of times that the raw observations are differenced), and the order of moving average (size of moving average window), respectively.

ARIMA works only with stationary time series. A stationary time series is one whose properties do not depend on the time at which the series is observed. One way to more objectively determine whether differencing is required is to use a unit root test. These are statistical hypothesis tests of stationarity that were designed for determining whether differencing is required. For this purpose, the Dickey–Fuller test was used (Table 2). The results of the test for output variables BEE_OUT and BEE_IN are presented below.

Table 2. Results of Dickey–Fuller test.

BEE_OUT		**BEE_IN**	
Results of Dickey–Fuller Test:		**Results of Dickey–Fuller Test:**	
Test Statistic	−8.410	Test Statistic	−9.169
p-value	2.112×10^{-13}	*p*-value	2.406×10^{-15}
#Lags Used	14	#Lags Used	3
Number of Observations Used	465	Number of Observations Used	476
Critical Value (1%)	−3.444	Critical Value (1%)	−3.444
Critical Value (5%)	−2.867	Critical Value (5%)	−2.867
Critical Value (10%)	−2.570	Critical Value (10%)	−2.570

We could overwhelmingly reject the null hypothesis of a unit root at all common significance levels. In other words, the observed time series were stationary.

6.2. Facebook Prophet

While ARIMA is autoregressive forecasting that fits a linear regression line with the lag values and error terms, Facebook Prophet is a procedure for forecasting time-series data on the basis of an additive model where nonlinear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well. This is based on generalized additive models (GAMs), which provide a general framework for extending a standard linear model by allowing for nonlinear functions of each of the variables while maintaining additivity. Just like linear models, GAMs can be applied with both quantitative and qualitative responses.

In this model, three main components were used: trend, seasonality, and holidays. They were combined in the following equation.

$$y(t) = g(t) + s(t) + h(t) + t, \tag{1}$$

where $g(t)$ is the trend function that models nonperiodic changes in the value of the time series, $s(t)$ represents periodic changes (e.g., weekly and yearly seasonality), and $h(t)$ represents the effects of holidays that occur on potentially irregular schedules over one or more days. Error term t represents any idiosyncratic changes not accommodated by the model. The detected components for the entire BEE_OUT time series, trend, daily behaviour, and the influence of the added regressors are shown in Figure 8. Similar graphics were obtained for the entire BEE_IN time series (Figure 9).

Facebook Prophet is very popular in time-series forecasting because it is robust to outliers, missing data, and dramatic changes in time series, whereas ARIMA is prone to white noise and nonstationary signals. The existence of outliers and missing data in such use cases is certain, bearing in mind that equipment may sometimes break down.

Here, we explore the problem of flexibly predicting Y on the basis of several predictors, $X1, \ldots, Xp$. Possible input variables were carefully selected from Table 1. More information of the selected features is provided in the results section.

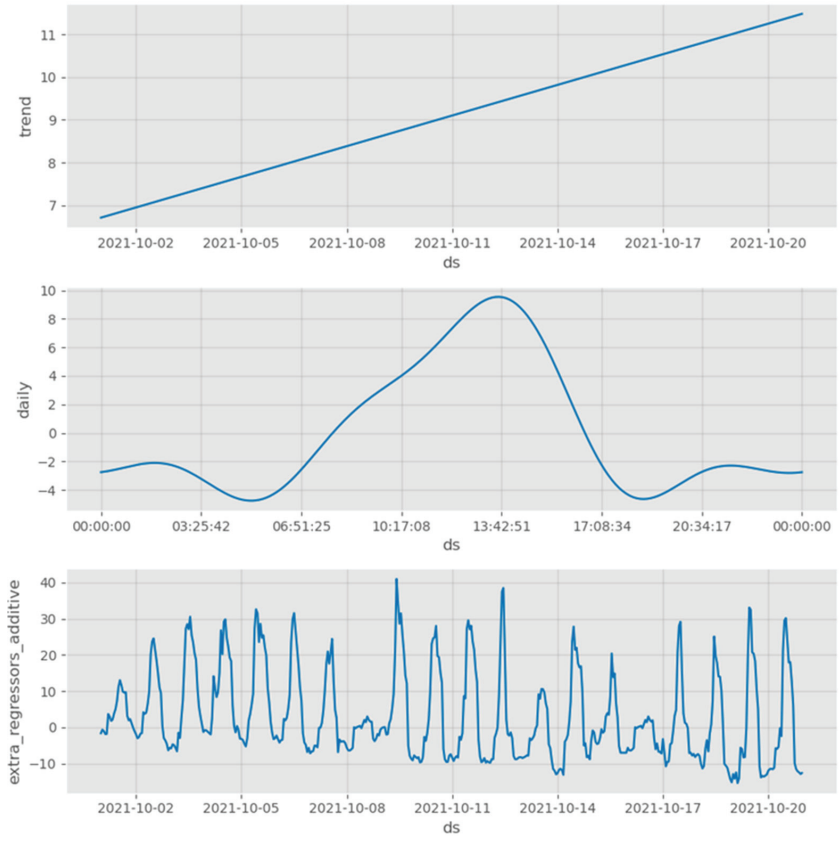


Figure 8. Results of training phase of Facebook Prophet algorithm for BEE_OUT variable.

6.3. LSTM Model

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows for it to exhibit temporal dynamic behavior. They are distinguished by their memory, as they take information from prior inputs to influence the current input and output. While traditional deep neural networks assume that inputs and outputs are independent of each other, the output of recurrent neural networks depends on prior elements within the sequence. While future events would also be helpful in determining the output of a given sequence, unidirectional recurrent neural networks cannot account for these events in their predictions.

These deep-learning algorithms are commonly used for ordinal or temporal problems such as language translation [73], natural language processing (NLP) [74], speech recognition [75], and image captioning [76].

There are three types of vanilla recurrent neural network: simple (RNN), gated recurrent unit (GRU), and long short-term memory unit (LSTM). The difference among them is shown in Figure 10, but we omit the details because they are outside the scope of this paper. Long short-term memory (LSTM) networks were invented by Hochreiter and Schmidhuber in 1997 [72], and they set accuracy records in multiple application domains. Here, LSTM cells were used for the time-series modeling.

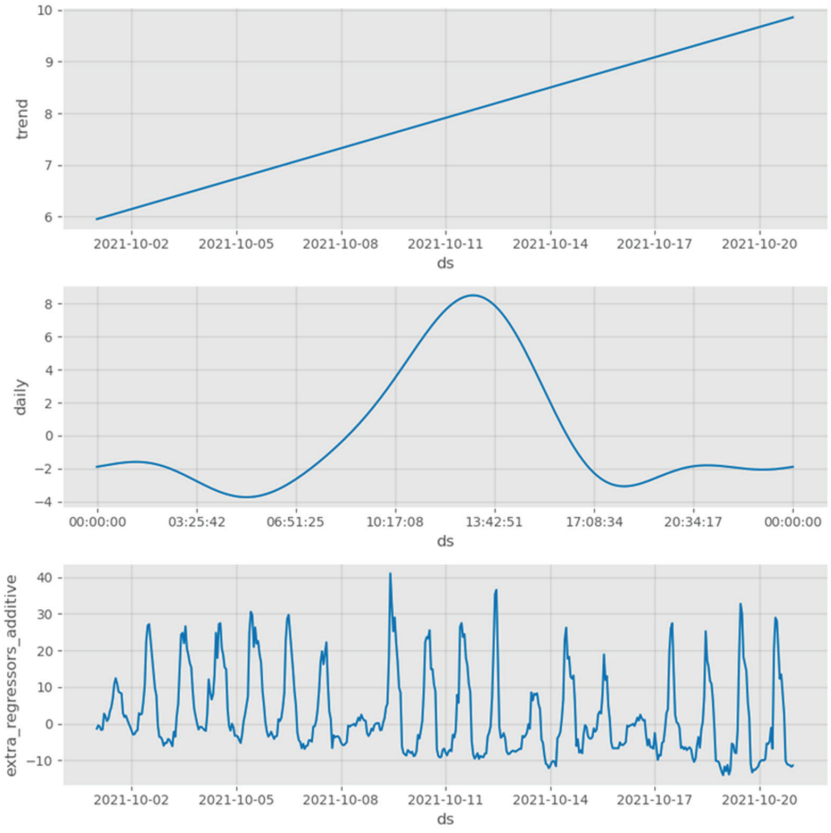


Figure 9. Results of training phase of Facebook Prophet algorithm for BEE_IN variable.

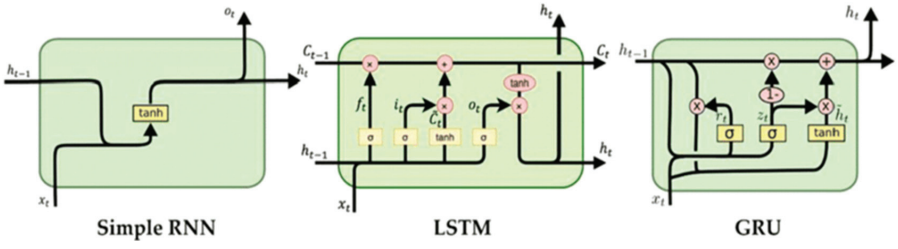


Figure 10. Network structure of RNN, LSTM, and GRU.

7. Experimental Setup and Evaluation

In order to test the robustness of the models, a time-series cross-validator was used in the experiments. The TimeSeriesSplit class from scikit-learn library provides a very simple interface to split time-series data samples that are observed at fixed time intervals into training and test sets. In each split, test indices are higher than before; thus, shuffling in the cross-validator is inappropriate. In other words, this cross-validation object is a variation of Kfold, where in the k-th split, it returns the first k folds as the training set, and the (k+1)-th fold as the test set.

a. ARIMA: In our experiments, different values for the p, d, and q parameters were tested, and the ARIMA model with the smallest RMSE error was selected for further

testing. For p, parameter values of 0, 1, 2, 4, 6, 8, and 10 were tested, while d and q values were tested for values ranging from 0 to 3. A combination of parameters (p, d, q) that showed the best performance of the ARIMA model for BEE_OUT and BEE_IN outputs was (p, d, q) = (10, 0, 2); for BEE_IN, the combination of (0, 0, 2) was selected.

b. Facebook Prophet: Different combinations of input variables from Table 1 were tested, but the best results were obtained by using the following variables: AM2302_1_Temp, AM2302_1_Humi, AM2302_2_Temp HIVE, AM2302_2_Humi HIVE, MHRD_rain, MQ135_PPM, BH1750_lux, VEML6750_uvindex, and Day_night.

Parameters with the greatest influence on movements used to produce the prediction model were temperature and relative humidity inside and outside of the hive, the presence of rain, air quality, the range and intensity of daylight, UV radiation, and night and day shifts.

The forecast for the entire BEE_OUT time series is shown in Figure 11. This figure is given only to show that Facebook Prophet can successfully learn from the observed time series. In the results, the complete time-series forecast and presented metrics are based on previously invisible data (test dataset).

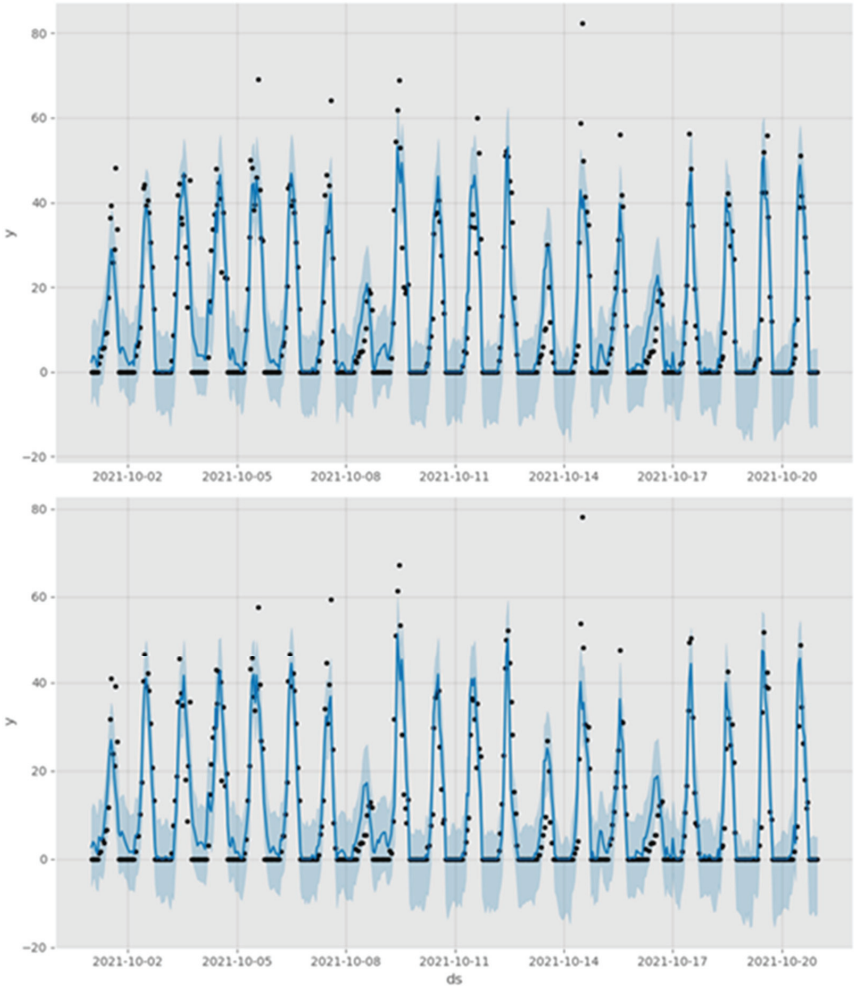


Figure 11. Forecast for the entire (20 days) BEE_OUT and BEE_IN time series.

a. Recurrent neural networks: The first step is to prepare the BEE dataset for the LSTM. This involves framing the dataset as a supervised-learning problem and normalizing the input variables. The same variables used by the Facebook Prophet algorithm were also used here. The supervised-learning problem is framed as predicting the bee exit or entrance at the current hour (t) given the bee exit or entrance measurement, and weather conditions at the prior time step. After this transformation step, the ten input variables (input series) and one output variable (bee exit or entrance at the current hour) are

$$var1(t-1),\ var1(t-1),\ \ldots,\ var10(t-1),\ var1(t) \tag{2}$$

We defined the LSTM with 50 neurons in the first hidden layer, and 1 neuron in the output layer for predicting bee activity. The input shape was one time step with 10 features. Mean absolute error (MAE) was used as the loss function and the efficient Adam version of stochastic gradient descent. The model was fit for 50 training epochs with a batch size of 20. Lastly, we monitored both training and test loss during the training phase. At the end of the run, both training and test loss were plotted. Resulting loss curves during the training and validation phases for the BEE_OUT and BEE_IN outputs are shown in Figures 12 and 13, respectively.

Separate time-series forecasts on the test set for each fold are shown in Figure 14. The machine learning (ML) applied to the time-series data, in this case, recurrent neural networks, is an efficient and effective way to analyze the data, apply a forecasting algorithm, and derive an accurate forecast.

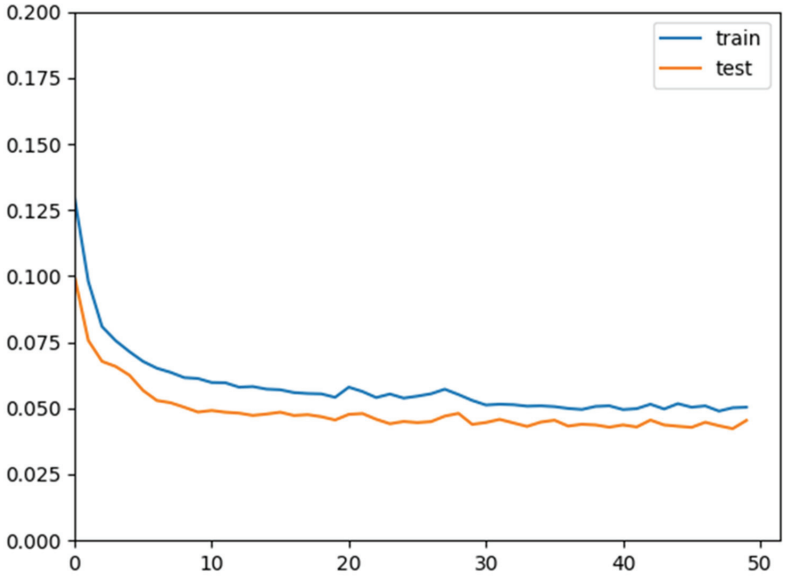


Figure 12. Bee OUT training phase.

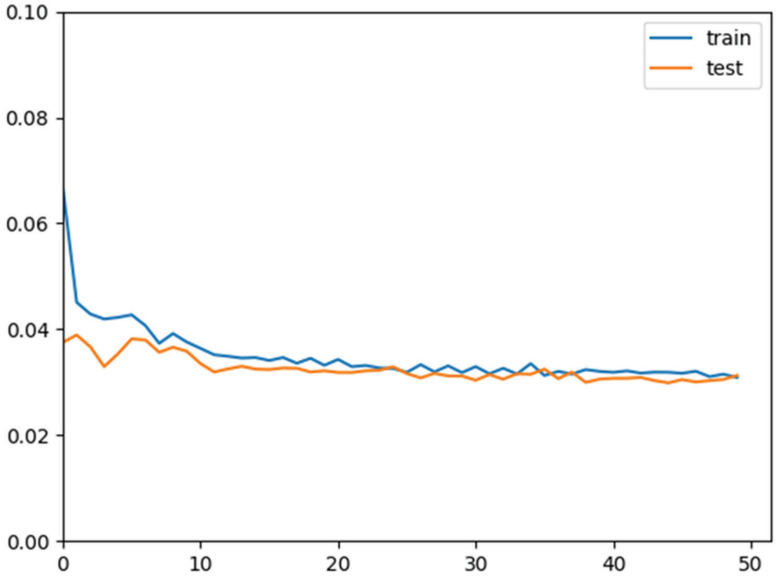


Figure 13. Bee IN training phase.

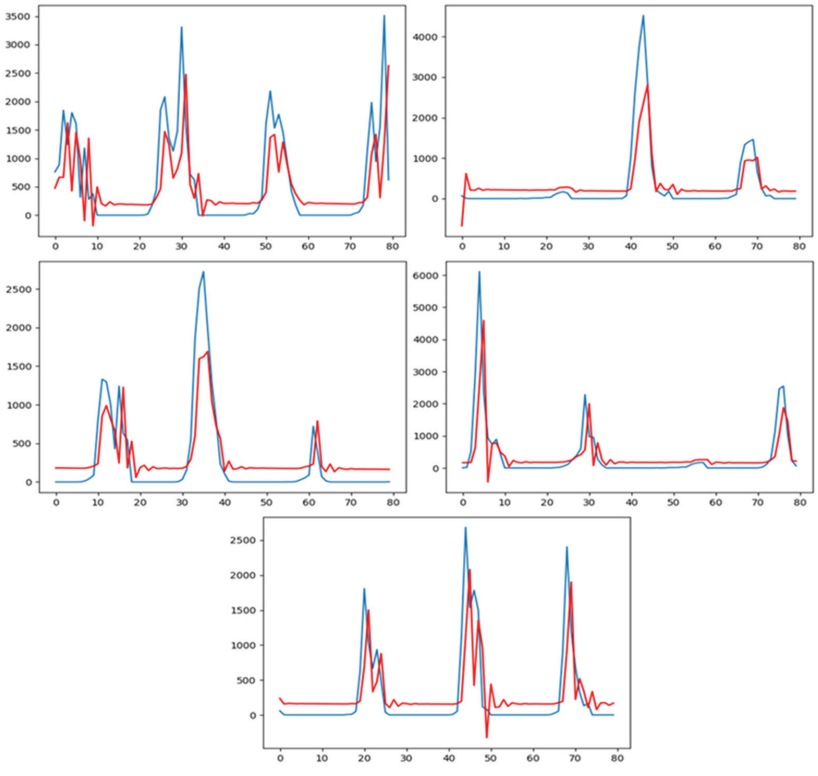


Figure 14. (blue) Original segment of BEE_OUT time series; (red) prediction for each fold (fivefold time-series split).

All aggregated results are shown in Table 3. The best results were achieved by using recurrent neural networks, where the average RMSE on the test sets was 426.49 for BEE_OUT time series; for the BEE_IN time series, RMSE had a value of 378.464.

Table 3. Results using recurrent neural network.

MODEL	CV Test RMSE OUT	CV Test RMSE IN
ARIMA	894.92	511.77
Facebook Prophet	589.97	475.25
LSTM	426.49	378.464

Table 4 shows the summarized optimal parameters for all investigated methods.

Table 4. Comparison of summarized optimal parameters for all investigated methods.

Algorithm Parameters	BEE OUT/IN
ARIMA	For BEE OUT (p, d, q): (10, 0, 2) For BEE IN (p, d, q): (0, 0, 2)
Facebook Prophet	Yearly seasonality: false Weekly seasonality: false Daily seasonality: true
Recurrent Neural Networks	RNN cell type: LSTM LSTM number: 50 Loss function: mean absolute error Batch size: 20 Optimizer: Adam Learning rate: 1×10^{-3} Epochs: 50

8. Conclusions

Comparisons of the experimental data against the model showed that our model represents the observed processes well. This is indicated by the results shown in the figures. According to the obtained results, the best model could achieve reliable bee activity prediction, with an error of only 8.9 missed bees per hour for bee exits from, and 7.8 missed bees per hour for bee entrances in a hive. We expect to see higher errors per hour when measurements are produced in the spring and summer months, and that additional feature engineering can help in model improving.

Apiculture presents complex problems pertinent to the life and wellbeing of bees. This paper presented a complete system for the monitoring and predictive analysis of honeybee activity, which addresses complex problems arising in beekeeping. Our aim was to improve existing solutions and create a fully developed system that would address some existing shortcomings.

The presented system is based on the application of IoT data collection and monitoring, machine-learning algorithms for beehive activity prediction, and remote control via IoT that enables undertaking certain corrective actions inside hives.

The increased number of sensors in the presented system is an important improvement over existing solutions. Each individual parameter influences bees in a different way and amount; however, when observed together and simultaneously, they provide more complete insight in the analysis of the results.

The application of advanced MAP enables the detection of sudden deviations and disruptions to the normal life of bees, and the prediction of potential disturbing changes. We showed that, by applying advanced algorithms, high-precision predictions on a daily

basis are possible. In this way, by employing a real-time monitoring application and push notifications of potential changes, the beekeeper has real-time insight into the conditions of the hives, and can react adequately to prevent unwanted outcomes.

There are some limitations to our approach. For example, the testing phase was conducted on two beehives, and the main data were collected from one hive that had not been moved during the experiment. The experiment was conducted during a period in which there was no food from flowers, and when bee activity was less than that during spring.

In future work, the system will be upgraded with appropriate weight sensors, oxygen/carbon dioxide sensors, thermal sensors, automatic bee-feeding, ventilation, and gate-closing systems, and connectivity with other applications and solutions.

In future papers, we will provide extensive research that includes analysis of the influence of microwaves and the presence of electronic components. It is also necessary to include time as a special factor in reaching conclusions because, from a longer time instance, we come to experiential conclusions, since every change, measurement, or analysis requires some time to pass for the results to be qualitative.

Author Contributions: Conceptualization, N.A. and V.U.; methodology, V.U. and B.A.; software, N.A. and D.H.; formal analysis, V.U. and B.S.; resources, N.A.; data curation, N.A., B.A. and D.H.; writing—original draft preparation, N.A. and B.A.; writing—review and editing, V.U., D.H. and B.S. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by [Higher Education Technical School of Professional Studies].

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

References

1. Ngo, T.-N.; Rustia, D.; Yang, E.-C.; Lin, T.-T. Honey Bee Colony Population Daily Loss Rate Forecasting and an Early Warning Method Using Temporal Convolutional Networks. *Sensors* **2021**, *21*, 3900. [CrossRef]
2. Hristov, P.; Shumkova, R.; Palova, N.; Neov, B. Factors Associated with Honey Bee Colony Losses: A Mini-Review. *Vet. Sci.* **2020**, *7*, 166. [CrossRef]
3. Watson, K.; Stallins, J.A. Honey Bees and Colony Collapse Disorder: A Pluralistic Reframing. *Geogr. Compass* **2016**, *10*, 222–236. [CrossRef]
4. Braga, A.R.; Gomes, D.G.; Rogers, R.; Hassler, E.E.; Freitas, B.M.; Cazier, J.A. A method for mining combined data from in-hive sensors, weather and apiary inspections to forecast the health status of honey bee colonies. *Comput. Electron. Agric.* **2020**, *169*, 105161. [CrossRef]
5. Zabasta, A.; Zhiravetska, A.; Kunicina, N.; Kondratjevs, K. Technical Implementation of IoT Concept for Bee Colony Monitoring. In Proceedings of the 2019 8th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 10–14 June 2019. [CrossRef]
6. Muhammad, Z.; Saxena, N.; Qureshi, I.M.; Ahn, C.W. Hybrid Artificial Bee Colony Algorithm for an Energy Efficient Internet of Things based on Wireless Sensor Network. *IETE Tech. Rev.* **2017**, *34*, 39–51. [CrossRef]
7. Edwards-Murphy, F.; Magno, M.; Whelan, P.M.; O'Halloran, J.; Popovici, E.M. b+WSN: Smart beehive with preliminary decision tree analysis for agriculture and honey bee health monitoring. *Comput. Electron. Agric.* **2016**, *124*, 211–219. [CrossRef]
8. Clarke, D.; Robert, D. Predictive modelling of honey bee foraging activity using local weather conditions. *Apidologie* **2018**, *49*, 386–396. [CrossRef]
9. Fiedler, S.; Zacepins, A.; Kviesis, A.; Komasilovs, V.; Wakjira, K.; Nawawi, M.; Hensel, O.; Purnomo, D. Implementation of the Precision Beekeeping System for Bee Colony Monitoring in Indonesia and Ethiopia. In Proceedings of the 2020 21th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, 27–29 October 2020.
10. Komasilovs, V.; Zacepins, A.; Kviesis, A.; Fiedler, S.; Kirchner, S. Modular sensory hardware and data processing solution for implementation of the precision beekeeping. *Agron. Res.* **2019**, *17*, 509–517. [CrossRef]
11. Catania, P.; Vallone, M. Design of an innovative system for precision beekeeping. In Proceedings of the 2019 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Portici, Italy, 24–26 October 2019; pp. 323–327.
12. Zacepins, A.; Brusbardis, V.; Meitalovs, J.; and Stalidzans, E. Challenges in the development of precision beekeep-ing. *Biosyst. Eng.* **2015**, *130*, 60–71. [CrossRef]
13. Catania, P.; Vallone, M. Application of A Precision Apiculture System to Monitor Honey Daily Production. *Sensors* **2020**, *20*, 2012. [CrossRef]
14. Henry, E.; Adamchuk, V.; Stanhope, T.; Buddle, C.; Rindlaub, N. Precision apiculture: Development of a wireless sensor network for honeybee hives. *Comput. Electron. Agric.* **2018**, *156*, 138–144. [CrossRef]
15. Kviesis, A.; Komasilovs, V.; Komasilova, O.; Zacepins, A. Application of fuzzy logic for honey bee colony state detection based on temperature data. *Biosyst. Eng.* **2020**, *193*, 90–100. [CrossRef]

16. Abou-Shaara, H.F.; Owayss, A.A.; Ibrahim, Y.Y.; Basuny, N.K. A review of impacts of temperature and relative humidity on various activities of honey bees. *Insectes Sociaux* **2017**, *64*, 455–463. [CrossRef]
17. Meikle, W.G.; Weiss, M.; Maes, P.W.; Fitz, W.; Snyder, L.A.; Sheehan, T.; Mott, B.M.; Anderson, K.E. Internal hive temperature as a means of monitoring honey bee colony health in a migratory beekeeping operation before and during winter. *Apidologie* **2017**, *48*, 666–680. [CrossRef]
18. Zacepins, A. Application of bee hive temperature measurements for recognition of bee colony state. In Proceedings of the 5th International Scientific Conference on Applied Information and Communication Technologies, Jelgava, Latvia, 26–27 April 2012; pp. 216–221.
19. Rybin, V.G.; Rodionova, E.A.; Karimov, A.I.; Kopets, E.E.; Chernetskiy, E.S. Remote Data Acquisition System for Apiary Monitoring. In Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), St. Petersburg, Russia, 26–29 January 2021; pp. 1059–1062.
20. Du, N.H.; Dong, N.D.; Luu, V.T.; Van Hoang, N.; Thai, P.H.; Ngoc, P.T.; Long, N.V.; Hong, P.T.T. Toward Audio Beehive Monitoring Based on IoT-AI techniques: A Survey and Perspective. *Vietnam J. Agric. Sci.* **2020**, *3*, 530–540. [CrossRef]
21. Craig, L.M.; Parry, R.M.; Tashakkori, R.; Watts, I. BeePhon: A Web-Application for Beehive Audio Exploration. In Proceedings of the 2019 SoutheastCon, Huntsville, AL, USA, 11–14 April 2019. [CrossRef]
22. Cecchi, S.; Terenzi, A.; Orcioni, S.; Riolo, P.; Ruschioni, S.; Isidoro, N. A Preliminary Study of Sounds Emitted by Honey Bees in a Beehive. In *Audio Engineering Society Convention 144*; Audio Engineering Society: Milan, Italy, 2018.
23. Murphy, F.E.; Srbinovski, B.; Magno, M.; Popovici, E.M.; Whelan, P.M. An automatic, wireless audio recording node for analysis of beehives. In Proceedings of the 2015 26th Irish Signals and Systems Conference (ISSC), Carlow, Ireland, 24–25 June 2015; pp. 1–6.
24. Kulyukin, V.; Mukherjee, S.; Amlathe, P. Toward Audio Beehive Monitoring: Deep Learning vs. Standard Machine Learning in Classifying Beehive Audio Samples. *Appl. Sci.* **2018**, *8*, 1573. [CrossRef]
25. Anand, N.; Raj, V.B.; Ullas, M.S.; Srivastava, A. Swarm Detection and Beehive Monitoring System using Auditory and Microclimatic Analysis. In Proceedings of the 2018 3rd International Conference on Circuits, Control, Communication and Computing (I4C), Bangalore, India, 3–5 October 2018; pp. 1–4.
26. Robles-Guerrero, A.; Saucedo-Anaya, T.; González-Ramérez, E.; Galván-Tejada, C.E. Frequency Analysis of Honey Bee Buzz for Automatic Recognition of Health Status: A Preliminary Study. *Res. Comput. Sci.* **2017**, *142*, 89–98. [CrossRef]
27. Szczurek, A.; Maciejewska, M.; Bąk, B.; Wilde, J.; Siuda, M. Semiconductor gas sensor as a detector of Varroa destructor infestation of honey bee colonies—Statistical evaluation. *Comput. Electron. Agric.* **2019**, *162*, 405–411. [CrossRef]
28. Bromenshenk, J.J.; Henderson, C.B.; Seccomb, R.A.; Welch, P.M.; Debnam, S.E.; Firth, D.R. Bees as Biosensors: Chemosensory Ability, Honey Bee Monitoring Systems, and Emergent Sensor Technologies Derived from the Pollinator Syndrome. *Biosensors* **2015**, *5*, 678–711. [CrossRef]
29. Hennessy, G.; Harris, C.; Eaton, C.; Wright, P.; Jackson, E.; Goulson, D.; Ratnieks, F.F. Gone with the wind: Effects of wind on honey bee visit rate and foraging behaviour. *Anim. Behav.* **2020**, *161*, 23–31. [CrossRef]
30. da Silva, D.; Rodrigues, Í.; Braga, A.; Nobre, J.; Freitas, B.; Gomes, D. An Autonomic, Adaptive and High-Precision Statistical Model to Determine Bee Colonies Well-Being Scenarios. In *Anais do XI Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais*; SBC: Porto Alegre, Brazil, 2020. [CrossRef]
31. Rybin, V.G.; Butusov, D.N.; Karimov, T.I.; Belkin, D.A.; Kozak, M.N. Embedded data acquisition system for beehive monitoring. In Proceedings of the 2017 IEEE II International Conference on Control in Technical Systems (CTS), Saint Petersburg, Russia, 25–27 October 2017; pp. 387–390.
32. Cecchi, S.; Spinsante, S.; Terenzi, A.; Orcioni, S. A Smart Sensor-Based Measurement System for Advanced Bee Hive Monitoring. *Sensors* **2020**, *20*, 2726. [CrossRef] [PubMed]
33. Murphy, F.E.; Magno, M.; Whelan, P.; Vici, E.P. b+WSN: Smart beehive for agriculture, environmental, and honey bee health monitoring—Preliminary results and analysis. In Proceedings of the 2015 IEEE Sensors Applications Symposium (SAS), Zadar, Croatia, 13–15 April 2015; pp. 1–6.
34. Meikle, W.G.; Holst, N. Application of continuous monitoring of honeybee colonies. *Apidologie* **2014**, *46*, 10–22. [CrossRef]
35. Marchal, P.; Buatois, A.; Kraus, S.; Klein, S.; Gómez-Moracho, T.; Lihoreau, M. Automated monitoring of bee behaviour using connected hives: Towards a computational apidology. *Apidologie* **2020**, *51*, 356–368. [CrossRef]
36. Ngo, T.N.; Wu, K.C.; Yang, E.-C.; Lin, T.T. A real-time imaging system for multiple honey bee tracking and activity moni-toring. *Comput. Electron. Agric.* **2019**, *163*, 104841. [CrossRef]
37. Cousin, P.; Cauia, E.; Siceanu, A.; de Cledat, J. The Development of an Efficient System to Monitor the Honeybee Colonies Depopulations. In Proceedings of the 2019 Global IoT Summit (GIoTS), Aarhus, Denmark, 17–21 June 2019.
38. Kridi, D.S.; de Carvalho, C.G.N.; Gomes, D.G. Application of wireless sensor networks for beehive monitoring and in-hive thermal patterns detection. *Comput. Electron. Agric.* **2016**, *127*, 221–235. [CrossRef]
39. Sachin, K.; Gagana, M.R.; Rubab, H.; Jalaja, G.S.; Jayanand, J. Monitoring of Honey Bee Hiving System using Sensor Networks. *Int. J. Eng. Res. Technol.* **2020**, *9*, 527–530. [CrossRef]
40. Zacepins, A.; Kviesis, A.; Komasilovs, V.; Muhammad, F.R. Monitoring System for Remote Bee Colony State Detection. *Balt. J. Mod. Comput.* **2020**, *8*, 461–470. [CrossRef]

41. Gil-Lebrero, S.; Quiles-Latorre, F.J.; Ortiz-López, M.; Sánchez-Ruiz, V.; Gámiz-López, V.; Luna-Rodríguez, J.J. Honey bee colonies remote monitoring system. *Sensors* **2017**, *17*, 55. [CrossRef]
42. Jiang, J.-A.; Wang, C.-H.; Chen, C.-H.; Liao, M.-S.; Su, Y.-L.; Chen, W.-S.; Huang, C.-P.; Yang, E.-C. A WSN-based automatic monitoring system for the foraging behavior of honey bees and environmental factors of beehives. *Comput. Electron. Agric.* **2016**, *123*, 304–318. [CrossRef]
43. Vidrascu, M.G.; Svasta, P.M.; Vladescu, M. High reliability wireless sensor node for bee hive monitoring. In Proceedings of the 2016 IEEE 22nd International Symposium for Design and Technology in Electronic Packaging (SIITME), Oradea, Romania, 20–23 October 2016; pp. 134–138.
44. Chen, W.-S.; Wang, C.-H.; Jiang, J.-A.; Yang, E.-C. Develpment of a Monitoring System for Honeybee Activities. In Proceedings of the 2015 Minth International Conference on Sensing Technology, Auckland, New Zealand, 8–10 December 2015; pp. 745–750.
45. Cecchi, S.; Terenzi, A.; Orcioni, S.; Spinsante, S.; Primiani, V.M.; Moglie, F.; Ruschioni, S.; Mattei, C.; Riolo, P.; Isidoro, N. Multi-sensor platform for real time measurements of honey bee hive parameters. *IOP Conf. Series Earth Environ. Sci.* **2019**, *275*, 012016. [CrossRef]
46. Dogan, S.; Akbal, E.; Ozmen Koca, G.; Balta, A. Design of a remote Controlled Beehive for Improving Efficiency of Beekeeping Activities. In Proceedings of the 8th International Advanced Technologies Symposium (IATS'17), Elazig, Turkey, 19–22 October 2017; Firat Universites: Elazığ, Turkey, 2017; pp. 1084–1090.
47. Balta, A.; Dogan, S.; Ozmen Koca, G.; Akbal, E. Software Modeling of Remote Controlled Beehive Design. In Proceedings of the International Conference on Advances and Innovations in Engineering (ICAIE), Elazig, Turkey, 10–12 May 2017; pp. 1133–1137.
48. Giammarini, M.; Concettoni, E.; Zazzarini, C.C.; Orlandini, N.; Albanesi, M.; Cristalli, C. BeeHive Lab project-Sensorized hive for bee colonies life study. In Proceedings of the 2015 12th International Workshop on Intelligent Solutions in Embedded Systems (WISES), Ancona, Italy, 29–30 October 2015; pp. 121–125.
49. Hong, W.; Xu, B.; Chi, X.; Cui, X.; Yan, Y.; Li, T. Long-Term and Extensive Monitoring for Bee Colonies Based on Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 7148–7155. [CrossRef]
50. Ochoa, I.Z.; Gutiérrez, S.; Rodriguez, F. Internet of Things: Low Cost Monitoring BeeHive System using Wireless Sensor Network. In Proceedings of the 2019 IEEE International Conference on Engineering Veracruz, ICEV 2019, Boca del Rio, Veracruz, 14–17 October 2019; pp. 1–7. [CrossRef]
51. Kontogiannis, S. An Internet of Things-Based Low-Power Integrated Beekeeping Safety and Conditions Monitoring System. *Inventions* **2019**, *4*, 52. [CrossRef]
52. Pešović, U.; Marković, D.; Đurašević, S.; Ranđić, S. Remote monitoring of beehive activity. *Acta Agric. Serbica* **2019**, *24*, 157–165. [CrossRef]
53. Debauche, O.; El Moulat, M.; Mahmoudi, S.; Boukraa, S.; Manneback, P.; Lebeau, F. Web monitoring of bee health for re-searchers and beekeepers based on the internet of things. *Procedia Comput. Sci.* **2018**, *130*, 991–998. [CrossRef]
54. Dineva, K. Computer system using internet of things for monitoring of bee hives. *Int. Multidiscip. Sci. GeoConference SGEM* **2017**, *17*, 27. [CrossRef]
55. Lyu, X.; Zhang, S.; Wang, Q. Design of Intelligent Beehive System based on Internet of Things Technology. In Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019), Chongqing, China, 30–31 May 2019.
56. Zabasta, A.; Kunicina, N.; Kondratjevs, K.; Ribickis, L. IoT Approach Application for Development of Autonomous Beekeeping System. In Proceedings of the 2019 International Conference in Engineering Applications (ICEA), Sao Miguel, Portugal, 8–11 July 2019; pp. 1–6.
57. Vidrascu, M.G.; Svasta, P.M. Embedded software for IOT bee hive monitoring node. In Proceedings of the 2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME), Constanța, Romania, 26–29 October 2017; pp. 183–188.
58. Zacepins, A.; Kviesis, A.; Pecka, A.; Osadcuks, V. Development of Intrenet of Things concept for Precission bee-keeping. In Proceedings of the 2017 18th International Carpathian Control Conference (ICCC), Sinaia, Romania, 28–31 May 2017; pp. 28–31.
59. Kviesis, A.; Zacepins, A. Application of neural networks for honey bee colony state identification. In Proceedings of the 2016 17th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, 29 May–1 June 2016; pp. 413–417.
60. Zgank, A. IoT-Based Bee Swarm Activity Acoustic Classification Using Deep Neural Networks. *Sensors* **2021**, *21*, 676. [CrossRef]
61. Chen, C.; Yang, E.-C.; Jiang, J.-A.; Lin, T.-T. An imaging system for monitoring the in-and-out activity of honey bees. *Comput. Electron. Agric.* **2012**, *89*, 100–109. [CrossRef]
62. Bermig, S.; Odemer, R.; Gombert, A.; Frommberger, M.; Rosenquist, R.; Pistorius, J. Experimental validation of an electronic counting device to determine flight activity of honey bees (*Apis mellifera* L.). *J. Cultiv. Plants* **2020**, *72*, 132–140. [CrossRef]
63. Arnia: Remote Beehive Monitoring. Available online: https://www.arnia.co/ (accessed on 28 January 2022).
64. Beecheck. Available online: https://beecheck.org/ (accessed on 28 January 2022).
65. Bee Counter. Available online: https://www.beehivemonitoring.com/en/ (accessed on 28 January 2022).
66. Honey Bee Counter. Available online: https://www.instructables.com/Honey-Bee-Counter/ (accessed on 28 January 2022).
67. Fritzing Open-Source Software. Available online: https://fritzing.org (accessed on 28 January 2022).
68. Supandi; Arkan, F.; Gusa, R.F.; Jumnahdi, M.; Kurniawan, R. Design of system for setting the temperature and monitoring bees in and out the hive. *IOP Conf. Series: Earth Environ. Sci.* **2020**, *599*, 012050. [CrossRef]

69. Hewamalage, H.; Bergmeir, C.; Bandara, K. Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *Int. J. Forecast.* **2021**, *37*, 388–427. [CrossRef]
70. Box, G.E.P.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*, 5th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2015; ISBN 9781118674925.
71. Taylor, S.J.; Letham, B. Forecasting at scale. *Am. Stat.* **2018**, *72*, 37–45. [CrossRef]
72. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
73. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Dean, J. Google's neural machine transla-tion system: Bridging the gap between human and machine translation. *arXiv* **2016**, arXiv:1609.08144.
74. Otter, D.W.; Medina, J.R.; Kalita, J.K. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Trans. Neural Networks Learn. Syst.* **2021**, *32*, 604–624. [CrossRef]
75. Nassif, A.B.; Shahin, I.; Attili, I.; Azzeh, M.; Shaalan, K. Speech recognition using deep neural networks: A sys-tematic review. *IEEE Access* **2019**, *7*, 19143–19165. [CrossRef]
76. Sharma, H.; Agrahari, M.; Singh, S.K.; Firoj, M.; Mishra, R.K. Image captioning: A comprehensive survey. In Proceedings of the 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 28–29 February 2020; pp. 325–328.

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland
www.mdpi.com

MDPI Books Editorial Office
E-mail: books@mdpi.com
www.mdpi.com/books

Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-0378-1