electronics

# Applications of Computational Intelligence

Edited by

Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Printed Edition of the Special Issue Published in *Electronics*

www.mdpi.com/journal/electronics

MDPI

# Applications of Computational Intelligence

# Applications of Computational Intelligence

Editors

**Yue Wu**
**Kai Qin**
**Maoguo Gong**
**Qiguang Miao**

MDPI • Basel • Beijing • Wuhan • Barcelona • Belgrade • Manchester • Tokyo • Cluj • Tianjin

*Editors*

Yue Wu
Xidian University
China

Kai Qin
Swinburne University of Technology
Australia

Maoguo Gong
Xidian University
China

Qiguang Miao
Xidian University
China

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

This is a reprint of articles from the Special Issue published online in the open access journal *Electronics* (ISSN 2079-9292) (available at: https://www.mdpi.com/journal/electronics/special_issues/AOCI_electronics).

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

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

**ISBN 978-3-0365-7038-9 (Hbk)**
**ISBN 978-3-0365-7039-6 (PDF)**

# Contents

# About the Editors

**Yue Wu**

Yue Wu received the B.Eng. and Ph.D. degrees from Xidian University, Xi'an, China, in 2011 and 2016, respectively. Since 2016, he has been a Teacher with Xidian University. He is currently an Associate Professor, Doctoral Supervisor, and Deputy Director of the Institute of Computational Intelligence at Xidian University. He has published more than 100 papers in high-level international journals and conferences and applied for and authorized more than 30 patents. His research interests include computational intelligence and its applications. He is the Registered Chairman of BIC-TA 2016 and ECOLE 2017, the Publishing Vice-Chairman of the 6th CCF Big Data Academic Conference, the Chairman of the IEEE CCIS2021 Organizing Committee, a Senior Member of the Chinese Computer Federation, etc. He is Editorial Board Member for over six journals, including *Remote Sensing*, *Applied Sciences*, *Electronics*, *Mathematics*.

**Kai Qin**

Kai Qin received a B.Eng. degree from Southeast University, Nanjing, China, in 2001, and a Ph.D. degree from Nanyang Technology University, Singapore, in 2007. From 2007 to 2017, he was with the University of Waterloo, Waterloo, ON, Canada; INRIA Grenoble Rhône-Alpes, Montbonnot-Saint-Martin, France; and RMIT University, Melbourne, VIC, Australia. He joined the Swinburne University of Technology, Hawthorn, VIC, Australia, in 2017, where he is currently a Professor. He is currently the Director of Swinburne Intelligent Data Analytics Lab, the Deputy Director of Swinburne Space Technology and Industry Institute, and the Program Lead of Swinburne Data Science Research Institute. His major research interests include machine learning, evolutionary computation, computer vision, remote sensing, services computing, and pervasive computing. Dr. Qin was a recipient of the 2012 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award. He is currently the Chair of the IEEE Computational Intelligence Society (CIS) Neural Networks Technical Committee and the Vice-Chair of the IEEE CIS Emergent Technologies Task Force on "Multitask Learning and Multitask Optimization".

**Maoguo Gong**

Maoguo Gong received B.Eng. and Ph.D. degrees from Xidian University, Xi'an, China, in 2003 and 2009, respectively. Since 2006, he has been a Teacher with Xidian University. He was promoted to an Associate Professor and a Full Professor, in 2008 and 2010, respectively, with exceptive admission. His research interests are broadly in the area of computational intelligence, with applications to optimization, learning, data mining, and image understanding. Dr. Gong received the prestigious National Program for the support of Top-Notch Young Professionals from the Central Organization Department of China, the Excellent Young Scientist Foundation from the National Natural Science Foundation of China, and the New Century Excellent Talent in University from the Ministry of Education of China. He is an Associate Editor of the *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* and the *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*.

**Qiguang Miao**

Qiguang Miao received the M.Eng. and Doctor degrees in computer science from Xidian University, Shaanxi, China. Since 2006, he has been a Full Professor in School of Computer Science and Technology at Xidian University, Xi’an, China, and the Director of Machine Learning and Intelligent Image Processing laboratory (MLIP). His research interests include pattern recognition, machine learning, and malware behavior analysis. In 2012, he was selected as a member of the program for New Century Excellent Talents in University of China by the Ministry of Education (MOE). He is a Member of IEEE; Member of ACM; IEEE CS Society; Senior Member of China Computer Federation (CCF) and AC of CCF YOCSEF, an Executive Member of Artificial Intelligent and pattern Recognition Council of CCF, and Members of Editorial board for IOT, etc.

# Preface to ”Applications of Computational Intelligence”

Computational Intelligence (CI) is the theory, design, application, and development of biologically and linguistically motivated computational paradigms. CI mainly includes three parts: neural networks, fuzzy systems, and evolutionary computation. CI has been widely used to solve complex problems in various domains, such as image processing, point cloud processing, and classification. Due to the great advantages of CI in dealing with practical application problems, increasing numbers of researchers have focused on the theoretical research and application of CI in recent years. This book, consisting of 20 articles written by research experts on topics of interest, reports the latest research on the applications of computational intelligence. Many novel and interesting methods are introduced, which provide guiding significance for the further development of computational intelligence.

**Yue Wu, Kai Qin, Maoguo Gong, and Qiguang Miao**
*Editors*

electronics

MDPI

Article

# Semi-Supervised Gastrointestinal Stromal Tumor Detection via Self-Training

**Qi Yang [1,†], Ziran Cao [2,†], Yaling Jiang [3], Hanbo Sun [2], Xiaokang Gu [2], Fei Xie [4,5,*], Fei Miao [6,*] and Gang Gao [7]**

[1] Department of Neurology, National Center for Neurological Disorders, Huashan Hospital, Fudan University, Shanghai 200433, China
[2] College of Information Science and Technology, Northwest University, Xi'an 710127, China
[3] School of Computer Science and Technology, Xidian University, Xi'an 710071, China
[4] Frontier Cross Research Institute, Xidian University, Xi'an 710071, China
[5] Xi'an Key Laboratory of Human-Machine Integration and Control Technology for Intelligent Rehabilitation, Xijing University, Xi'an 710123, China
[6] Department of Ultrasound, Ruijin Hospital, Shanghai Jiao Tong University School of Medicine, Shanghai 200025, China
[7] Shanghai Yiran Health Consulting Co., Ltd., Shanghai 201821, China
* Correspondence: fxie@xidian.edu.cn (F.X.); mf11066@rjh.com.cn (F.M.)
† These authors contributed equally to this work.

**Abstract:** The clinical diagnosis of gastrointestinal stromal tumors (GISTs) requires time-consuming tumor localization by physicians, while automated detection of GIST can help physicians develop timely treatment plans. Existing GIST detection methods based on fully supervised deep learning require a large amount of labeled data for the model training, but the acquisition of labeled data is often time-consuming and labor-intensive, hindering the optimization of the model. However, the semi-supervised learning method can perform better than the fully supervised learning method with only a small amount of labeled data because of the full use of unlabeled data, which effectively compensates for the lack of labeled data. Therefore, we propose a semi-supervised gastrointestinal stromal tumor (GIST) detection method based on self-training using the new selection criterion to guarantee the quality of pseudo-labels and adding the pseudo-labeled data to the training set together with the labeled data after linear mixing. In addition, we introduce the improved Faster RCNN with the multiscale module and the feature enhancement module (FEM) for semi-supervised GIST detection. The multiscale module and the FEM can better fit the characteristics of GIST and obtain better detection results. The experiment results showed that our approach achieved the best performance on our GIST image dataset with the joint optimization of the self-training framework, the multiscale module, and the FEM.

**Keywords:** gastrointestinal stromal tumor; semi-supervised learning; self-training; object detection; computational intelligence

**Citation:** Yang, Q.; Cao, Z.; Jiang, Y.; Sun, H.; Gu, X.; Xie, F.; Miao, F.; Gao, G. Semi-Supervised Gastrointestinal Stromal Tumor Detection via Self-Training. *Electronics* **2023**, *12*, 904. https://doi.org/10.3390/electronics12040904

Academic Editor: Yu Zhang

Received: 4 January 2023
Revised: 31 January 2023
Accepted: 4 February 2023
Published: 10 February 2023

## 1. Introduction

A gastrointestinal stromal tumor (GIST) is a highly aggressive gastrointestinal mesenchymal tumor, mainly diagnosed with image examination. GIST detection on abdominal CT images helps acquire the location of tumors, formulate treatment plans in time, and prevent distant metastasis of tumors. Currently, computer vision technology based on the neural network has been widely used in a variety of lesion detection tasks, including the diagnosis of pulmonary tuberculosis, breast cancer, and other lesions. Unlike lesion detection in other organs, the small intestine has a motor function, so the shape of the GIST varies widely, leading to low discriminability and many difficulties for GIST detection. Therefore, there are few studies related to the detection and identification of GIST, and existing techniques still use common object detection algorithms (Faster R-CNN [1], YOLO [2], Cascade R-CNN [3], RetinaNet [4], etc.). These fully supervised learning methods heavily rely on labeled data for the parameter optimization, but the medical images must be labeled by doctors with clinical experience, which requires considerable resources. As a result, the detection results of neural networks are affected by the lack of labeled data. Meanwhile, a substantial number of unlabeled images are stored in the hospital's medical system. Compared with the high cost of manually labeling them, it is much easier to retrieve these images. To solve the problem of unsatisfactory training results caused by the imbalance between the number of labeled and unlabeled images, we use the semi-supervised learning method to train the model.

With limited labeled data, the semi-supervised learning (SSL) method can improve the performance of the model by effectively using a tremendous amount of unlabeled data to optimize the model while reducing the dependence on labeled data. Generally, the SSL method can be divided into two steps: (a) training on a small amount of labeled data to obtain model A and predicting pseudo-labels of unlabeled data through A; and (b) retraining the model on a new dataset consisting of pseudo-labeled and labeled data to improve the performance of the model. Since the pseudo-labels of the unlabeled data are generated with the model prediction, there may be some mistakes. Some researchers [5,6] have used self-integration methods to improve the quality of pseudo-labels and enhance the robustness of the model. In addition, there are also algorithms [7,8] that learn complementary information by cotraining to avoid confirmation bias and guarantee the accuracy of pseudo-labels. In order to prevent wrong pseudo-labels from producing errors that continue to iterate and affect the performance of the model, we propose a self-training-based SSL method (Figure 1) that uses the dual constraints of dynamic threshold and IOU to enhance the quality of pseudo-labels. The dynamic threshold constraint means setting a minimum threshold for the confidence of the pseudo-label, using a higher confidence threshold at the beginning of the training and gradually decreasing it as the training progresses. The IOU constraint means that the intersection over union (IOU) between multiple pseudo-labels of different transformed images should be greater than the set threshold; that is, after the data augmentation on the unlabeled image is completed, the shape and the position of the new candidate bounding box should maintain a certain immutability.

Additionally, to fundamentally ensure the quality of pseudo-labels, we improved the popular two-stage object detector (Faster R-CNN [1]) and applied the Improved Faster R-CNN to pseudo-label generation. Most of the traditional object detection algorithms used in existing GIST detection techniques are designed for object detection tasks in natural images and perform well on salient object detection. However, GIST has an unclear boundary and inconspicuous features in abdominal CT, and these algorithms cannot achieve good results. Compared with these techniques, Improved Faster R-CNN makes adjustments targeted to the morphological characteristics of GIST. In Improved Faster R-CNN, the multiscale module and the feature enhancement module (FEM) designed for the characteristics of GIST have been added. The newly added module can better detect GISTs of different scales in complex backgrounds, which helps to improve the accuracy of pseudo-labels. Finally, Mixup [9] can be used to augment the true labels of the labeled data and the pseudo-labels of the unlabeled data. The generalization capability of the network can be significantly enhanced by linearly mixing these samples.

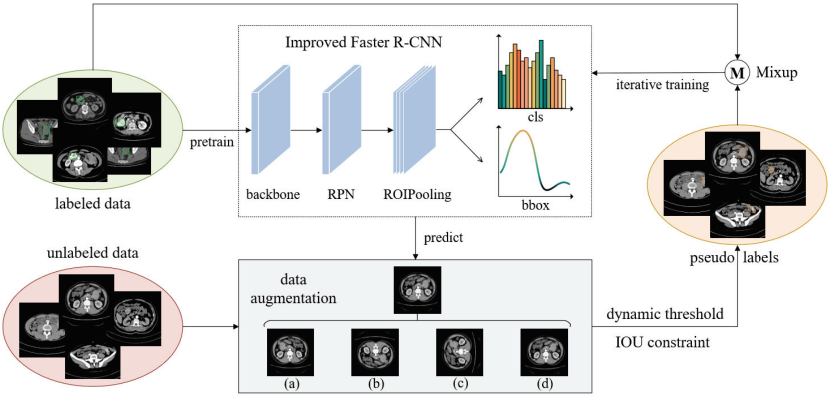


**Figure 1.** Overview of the self-training method. In each iteration, the predictions of the Improved Faster R-CNN on unlabeled data are augmented and then filtered out with the IOU constraint and the dynamic threshold together to generate pseudo-labels. The labeled data and the pseudo-labeled data are linearly mixed using Mixup, and the resulting new dataset is used for the next iteration. The data augmentation methods in the figure are (a) horizontal flip, (b) vertical flip, (c) rotation, and (d) affine transformation; one is randomly selected as the data augmentation method and repeated k times during the experiment.

In summary, our main contributions are as follows: (1) We propose a detection algorithm (Improved Faster R-CNN) for GIST detection and use it as the benchmark model for the SSL method. (2) We propose a novel self-training-based SSL method for GIST detection. (3) Extensive experiments demonstrated that the performance of the proposed SSL method is significantly improved compared to the fully supervised learning method.

## 2. Related Work

### 2.1. Lesion Detection

Lesion detection is an important computer vision task in the field of CAD (computer-aided diagnosis) and has received considerable attention in recent years. Many scholars have designed excellent object detectors based on convolutional neural networks (CNNs) for lesion detection. Cireşan et al. [10] added max-pooling layer and postprocessing strategies to the CNN for mitosis detection on mammary gland histological images. Setio et al. [11] proposed a multiview convolutional network that combines the respective advantages of three detectors for pulmonary nodule detection. Rajpurkar et al. [12] improved the dense convolutional network by replacing the fully connected layer with a single output layer and applying a nonlinear sigmoid activation function to achieve excellent performance on the task of pneumonia detection on chest radiographs. Sedik et al. [13] constructed a deep learning architecture for COVID-19 detection on CT images and X-ray films based on CNN and ConvLSTM, which included convolutional, pooling, and ConvLSTM layers, and the multilayered structure effectively reduced the overfitting errors and enhanced the detection accuracy. Although artificial intelligence technologies have been widely used in the field of CAD, there is still little research on GIST detection. The small intestine moves by its nature, and the GIST appears with considerable morphological differences in abdominal CT, resulting in difficulty in improving the accuracy of GIST detection. At present, only our team has carried out this research. Fei et al. [14] have combined a variety of classical fully supervised detection algorithms to improve the accuracy of GIST detection, but the theoretical innovation of this method needs to be improved. In addition, the method requires training on a large amount of accurate manually annotated data, which is expensive and time-consuming to acquire in the medical imaging domain. In contrast to previous methods that solely train models on labeled data, our SSL approach trains an object detector on both labeled and unlabeled data.

### *2.2. Semi-Supervised Object Detection*

Semi-supervised learning methods can leverage latent knowledge from unlabeled data to facilitate model learning with limited labeled data [15]. Existing SSL methods consist of two categories: consistency-based methods and self-training-based methods. The main idea of the consistency-based [16–18] approach is that for any input data, its output should be consistent with the original output when it is disturbed by less noise. Self-training-based approaches improve the performance of SSL by filtering noisy labels using a predefined threshold and adding the retained pseudo-labels into model retraining. Lee et al. [19] used the deep neural network to train both labeled and unlabeled data simultaneously and pioneered the method of using pseudo-labels for training. Iscen et al. [20] used a transduction label propagation method based on the prevalence hypothesis in predicting pseudo-labels and achieved transduction learning by calculating the similarity matrix between the labeled and unlabeled data. Qizhe Xie et al. [21] improved the quality of pseudo-labels through repeated teacher–student model iterations to enhance the robustness and accuracy of self-training. Considering the uncertainty of the teacher network in the self-training method, Mukherjee et al. [22] chose the Bayesian network to estimate the uncertainty of pseudo-labels, thereby reducing the influence of noisy labels on the model.

The SSL method has also been widely used in the object detection field, and many researchers are committed to training high-performance object detectors with a limited amount of labeled data and a large amount of unlabeled data. Jeong et al. [23] proposed the CSD method based on consistency regularization, which calculates the consistency loss between the prediction on the original unlabeled image and the flipped unlabeled image to achieve the aim of fully utilizing unlabeled data. Sohn et al. [24] combined both self-training and consistency regularization to propose the STAC method, which first eliminates some low-confidence pseudo-labels obtained from self-training by threshold screening and calculates unsupervised loss as well as a supervised loss while training on the augmented unlabeled data together with labeled data. Qize et al. [25] performed self-training object detection based on the mean teacher model, using the nonmaximum suppression (NMS) method to fuse the detection results from different iteration periods to ensure the stability of the detection results during the training process. Moreover, the use of double-head can effectively utilize the complementary information and improve the quality of the pseudo-labels. To address the problem of the imbalance between the foreground and background, Fangyuan et al. [26] proposed a self-training method of adaptive class rebalancing that stores and extracts foreground instances and pastes them into random positions of training samples, increasing the proportion of foreground instances. They also designed a two-stage filter to weed out unreliable pseudo labels.

Although the SSL method for object detection has garnered a degree of success, the following problems in SSL-based GIST detection still remain: (1) The current semi-supervised object detection methods only employ basic object detection algorithms, such as Faster R-CNN [1]. However, given that GIST does not have clear boundary features in the CT image and has a large degree of variation, it is necessary to use an object detection algorithm that is more suitable for these characteristics. (2) The current object detection approach uses self-training without taking into account erroneous pseudo-labels, which results in overfitting to the wrong pseudo-labels and a decrease in the model's accuracy. As a result, we have built our SSL framework using the Improved Faster R-CNN with the multiscale module and the FEM. We then added the dynamic threshold and the IOU constraint in the self-training process to increase the accuracy of pseudo-labels, ensuring that the following model iterations perform better.

## 3. Method

With a severe lack of labeled data, using limited labeled data to improve model performance has become a significant problem in GIST detection. To fully utilize all data, including unlabeled data, we propose an SSL method based on self-training and design Improved Faster R-CNN as the detection algorithm according to the characteristics of GIST. The Improved Faster R-CNN containing the multiscale module and the FEM can better integrate multidimensional feature information and combine deep semantic information with shallow location information. We further developed a new pseudo-label selection strategy to improve the robustness of the model. By applying the dynamic threshold constraint and the IOU constraint to the prediction results of unlabeled data, the reliable pseudo-labels can be retained for subsequent training. In the subsequent sections, we describe improvements to the Faster R-CNN by introducing two new modules aimed at characterizing GISTs (Section 3.1). Next, we introduce a novel pseudo-label selection strategy (Section 3.2) and outline our self-training approach (Section 3.3).

### *3.1. Improved Faster R-CNN*

We optimized the Faster R-CNN to improve the accuracy of GIST detection and ensure the quality of pseudo-labels; the network structure is shown in Figure 2. In this paper, two optimization modules are proposed: (1) Given the large variability of GISTs, a multiscale module was developed to use feature information of various levels. (2) The FEM was introduced to combine channel and spatial dimension information for the complex background of GIST images.

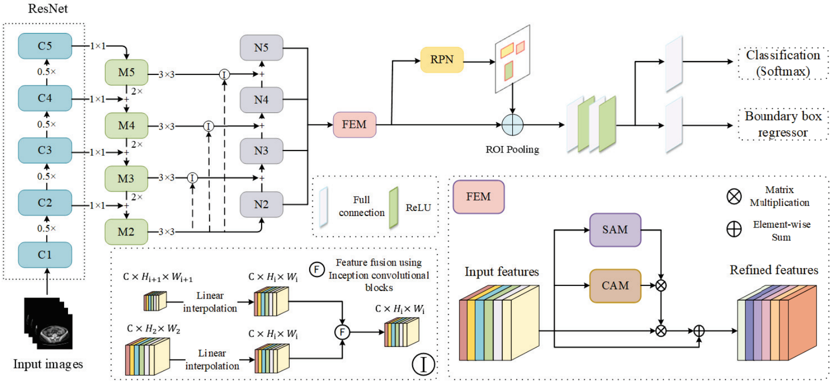


**Figure 2.** The architecture of the Improved Faster R-CNN  which consists of several main components: ResNet, the multiscale module, FEM, RPN, and ROI. The input images are processed using ResNet to obtain the feature maps of each layer, and the results are combined using the multiscale module to feed into the FEM. The feature maps enhanced with the FEM are fed into the RPN for anchor proposal, and the ROI then collects the input feature maps and proposals to extract the proposal feature maps into the full connected layer.

One of the challenges of GIST detection is that the object scale varies excessively. Using the single-layer feature map for prediction may affect the accuracy of the result due to the limited information, so the feature maps at different levels should be combined for detection. The traditional feature pyramid network (FPN) [27] can fuse information of the low-level with that of the high-level, but there are still some problems: (1) The transmission path between the low-level features and high-level features is too long, which increases the difficulty of access. (2) Although FPN utilizes the information of different layers, each layer only contains the information of the current layer and higher layers. The lack of location information of lower levels is not conducive to small target detection. In response to the problems in the FPN, we improve it by adding a bottom-up connection based on the original path. When downsampling the feature map of the $N_i$ layer, $M_2$ and $M_{i+1}$ are bilinearly interpolated to resize to the same scale (the size of the feature map of the $N_i$ layer), and then the fused results are combined with the feature map of the $N_i$ layer to obtain the feature map of the $N_{i+1}$ layer. We choose the Inception [28] convolution block for feature map fusion to solve the problem of excessive computation caused by a large convolution kernel. The improved FPN structure enables the feature map of each layer to contain both the semantic information of the deeper layers and the rich localization information of the first layer, assisting the model in performing better detection.

Another challenge of GIST detection is the difficulty of distinguishing the foreground from the background in CT images. The lesion area shares certain similarities with the surrounding background, and it is hard for the basic model to separate the object, so the FEM is introduced. The feature map obtained through convolution only contains the spatial information in the local receptive field and lacks the connection between each channel. If the information of each channel is only processed globally, the information interaction within the space is missed. Our FEM uses both a channel attention mechanism and a spatial attention mechanism to enhance feature representation, highlight relevant features of the GIST lesion area, and suppress background noise, thus enhancing the feature extraction ability of the network.

We use the channel attention mechanism (CAM) [29] to model the correlation between each channel and obtain the weight of each channel. The process can be written as follows:

$$MLP_1 = Conv_1^{1/r}(Relu(Conv_1^r(P_{\max}(F)))), \tag{1}$$

$$MLP_1 = Conv_1^{1/r}(Relu(Conv_1^r(P_{avg}(F)))), \tag{2}$$

$$CA(X) = \sigma(MLP_1 + MLP_2), \tag{3}$$

where $F$ represents the original feature map, $P_{\max}$ and $P_{avg}$ denote the max pooling and the average pooling, $Conv_1^{1/r}$ denotes that the convolution kernel size is $i \times i$ and the number of channels becomes $1/r$ times of the original, and $\sigma$ is the *Sigmoid*.

The spatial attention mechanism (SAM) [29] is used to model the correlation of the spatial position on the feature map of each channel and calculate its weight. The feature map is calculated as follows:

$$SA(F) = \sigma(Conv_7(P_{\max}(F) \| P_{avg}(F)), \tag{4}$$

where $Conv_7$ denotes that the convolution kernel size is $7 \times 7$, and $\|$ represents merging in the channel dimension.

The FEM refers to BAM [30] and establishes a parallel connection between the CAM and the SAM. Finally, the calculating process can be expressed as follows:

$$\begin{aligned} F'' &= FEM(F) = F + F \times F' \\ &= F + F \times \sigma(expand(CA(F)) \times expand(SA(F))). \end{aligned} \tag{5}$$

### 3.2. Pseudo-Label Selection Strategy

The correctness of pseudo-labels is crucial for subsequent training iterations. If incorrect pseudo-labels are added to the dataset, this will hinder the optimization of model parameters. To this end, we designed a pseudo-label selection strategy based on the dynamic threshold and the IOU constraint, which can effectively screen out pseudo-labels with a higher correct probability and help the model converge.

The method of selecting pseudo-labels by using an unchanging threshold has numerous drawbacks. If the threshold is set too high, the model will filter out the candidate bounding box in the target area and prevent it from being added to the pseudo-label set, leading to a large number of false negative examples in the subsequent training phase. In contrast, if the threshold is set too low, numerous candidate bounding boxes in the nontarget area will be added to the pseudo-label set, thus generating many false-positive examples in the next round of training. In fact, as training progresses, the network's detecting ability gradually advances, and the validity of the generated pseudo-labels rises. Therefore, the threshold value used to choose the pseudo-label should be dynamic. To avoid incorrect pseudo-labels from influencing model training, we set a high selection threshold at the early stage of training. As training proceeds, we gradually lower this threshold to prevent correct pseudo-labels from being eliminated. Selecting pseudo-labels through the dynamic threshold makes more sense. The value of threshold in the $q^{th}$ round is defined as follows:

$$T_q = \begin{cases} 0.95, q = 1 \\ T_{q-1} - (q-1) \times 0.05, q > 1 \end{cases} . \quad (6)$$

Based on the dynamic threshold, we created a new IOU constraint. The IOU constraint sets the condition for the retention of pseudo-labels. Only when the IOU between the detection results of various transformed images is higher than 0.9 do we regard the bounding box as the pseudo-label. Figure 3 shows the results after applying the IOU constraint. By comparing the original results with the true label, it can be found that the detection box on the right is a false-positive example, which will affect the optimization of the model if it is kept as a pseudo-label. After the IOU constraint is applied, the false-positive bounding box can be successfully excluded, which further ensures the quality of the pseudo labels.

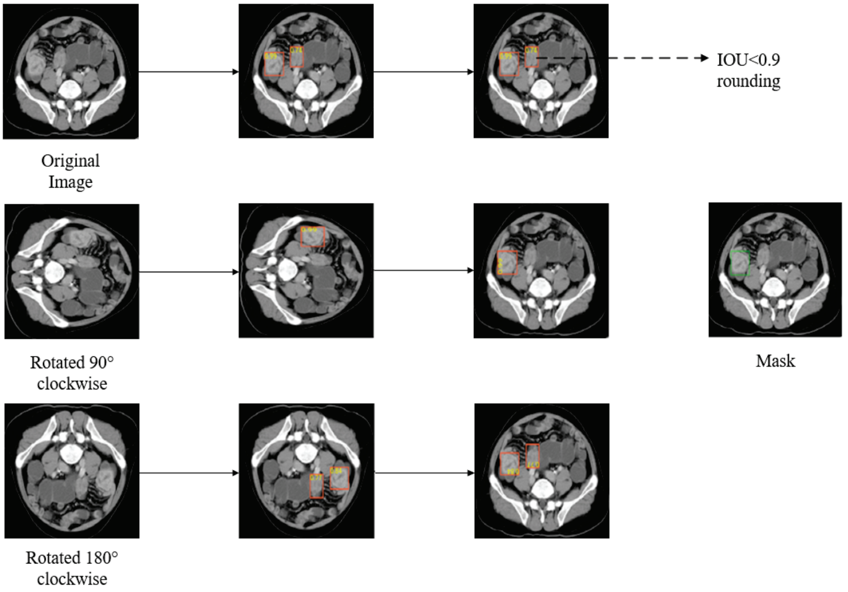


**Figure 3.** The effect of IOU constraint (the orange bounding box in the figure represents the predicted box, and the green bounding box represents the actual labeled box).

By synthesizing the dynamic threshold and the IOU constraint, the selecting criteria of pseudo-labels in the $q^{th}$ round can be expressed as follows:

$$P_i^j = \begin{cases} 1, \ f_i^j > T_q \quad and \quad IOU > 0.9 \\ 0, \ otherwise \end{cases} , \quad (7)$$

where i and j denotes the $j^{th}$ bounding box of the $i^{th}$ image, $f_i^j$ is the confidence of the bounding box, and $P_i^j = 1$ represents that the pseudo-label corresponding to this bounding box is retained, with the opposite being discarded.

### 3.3. Self-Training Method

In this paper, we propose a semi-supervised GIST detection algorithm based on the self-training method, which aims to improve the effectiveness of GIST detection with a small amount of labeled data and a large amount of unlabeled data. The whole procedure of the GIST detection is shown in Algorithm 1.

**Algorithm 1** Procedure of Semi-Supervised GIST Detection.

```
Input: Labeled data, $L$; Unlabeled data, $U$; Data augmentation strategies $T$ in $\{t_1, t_2, \ldots, t_m\}$
Output: Trainable parameters of network, $W$
Initialize hyperparameters: rounds of iteration $Q$, times of data augmentation $K$, threshold $P_1$
Pretrain the model on $L$ to get the initial parameters $W$
for $q = 1 : Q$ do
    for $k = 1 : K$ do
        Use $W$ to predict on $t_i(U)$
    end for
    $P_q \leftarrow P_q - (q-1) \times 0.05$
    Filter the results according to (7) to obtain the set of pseudo-labels
    Reassemble to acquire the new training set: $L_n = L \cup R \cup (Mixup(L, R))$
    Retrain the model on $L_n$ to acquire the new parameters $W$
end for
Return $W$
```

For the labeled data, the labels are the actual bounding boxes, and the confidence is set to 1. We first train with the labeled data to obtain the initial model and then apply different data augmentation strategies to the unlabeled data following the data distillation method proposed by Radosavovic et al. [31].

The data augmentation strategies used in this paper mainly include flip, rotation, and affine transformation. When the flip is chosen as the data augmentation method, the corresponding detection result needs to be flipped as well. For the affine transformation, the set translation range does not exceed 10 pixels, and the position of the bounding box does not vary greatly, so it can remain unchanged. For the rotation operation, the given rotation angle is an integer multiple of 90° or less than 10°. When the angle does not exceed 10°, the position of the bounding box stays unchanged, referring to the affine transformation operation; when the image is rotated 90° clockwise, the coordinates of the corresponding bounding box need to be rotated 90° counterclockwise, and so on for other angles.

The initial model detects the images after data augmentation 1 to k times respectively, and all the results are fused to obtain the pseudo-labels. The pseudo-labels generated by prediction may have some errors. For this reason, we use the dynamic threshold and the IOU constraint to enhance the quality of pseudo-labels. The dynamic threshold refers to a threshold that changes dynamically for the confidence of the pseudo-label, utilizing a higher confidence threshold in the early stages of training and progressively lowering the threshold as training progresses. The IOU constraint is a constraint on the overlap area between the bounding boxes predicted by the initial model on the images after data augmentation 1 to k times. After the transformation, the bounding box is used as a pseudo-label for that image only if it appears on all images with a similar position and size.

After clean pseudo-labels are filtered out through the above-mentioned constraint strategies, Mixup [9] is used to linearly mix the labeled data and the pseudo-labeled data. The new samples acquired after mixing are then used once more for the training, which can substantially enhance the network's generalization capacity.

Mixup is a crucial part of the MixMatch [16] framework, which enables the model to obtain better generalization performance by linearly interpolating pairwise training samples. The traditional Mixup is designed for image classification tasks, where each image is associated with one class label. The generated image $\widetilde{x}$ and its label $\widetilde{y}$ can be defined as follows:

$$\widehat{x} = \lambda x + (1-\lambda)x', \tag{8}$$

$$\widehat{y} = \lambda y + (1-\lambda)y', \tag{9}$$

where $x$ and $x'$ denote two different images, $y$ and $y'$, respectively, denote their probability of the corresponding class, $\lambda \in [0,1]$.

Since the data used in this paper are annotated with the bounding box of the lesion, we opted for image-level Mixup rather than classification Mixup. The generated label $\widehat{x}_i$ and its confidence $\widehat{y}_i$ in image-level Mixup can be calculated as follows:

$$\widetilde{\lambda} = \max(\lambda, 1-\lambda), \tag{10}$$

$$\widehat{x}_i = \widetilde{\lambda}x_i + \left(1-\widetilde{\lambda}\right)x'_i, \forall i, \tag{11}$$

$$\widehat{y}_i = \widetilde{\lambda}y_i + \left(1-\widetilde{\lambda}\right)y'_i, \forall i, \tag{12}$$

where $\widehat{x}_i$ denotes the $i^{th}$ label on the generated image $\widehat{x}$, $x$ and $x'$ denote bounding boxes on two different images, and $y$ and $y'$, respectively, represent their confidence, $\lambda \in [0,1]$.

## 4. Results

We first conducted ablation studies to verify the effectiveness of the proposed module and strategies. Furthermore, we designed experiments to demonstrate the superiority of the Improved Faster R-CNN and the self-training method. Detailed information on the configuration and results is presented in the following subsections.

### *4.1. Datasets and Experimental Settings*

Datasets used for the fully supervised method (Improved Faster R-CNN) were the following: The datasets used in the experiment were the CT images (a series of DICOM format files obtained by doctors using related equipment to scan the patient's abdomen) of GIST patients provided by the hospital, including labeled images of 213 patients and unlabeled images of 10 nonpatients. Each patient had between 50 and 80 slices, of which only 3 to 10 slices contained GIST. The slices with lesions labeled by qualified medical professionals were used as the datasets in this work. We used pydicom to convert DICOM format files to png format images and finally obtained a total of 3735 images with GIST annotated by doctors, comprising 526 ones with the small object, 2212 ones with the medium object, and 997 ones with the large object. Of the labeled images, 70% were used for training, and the remaining 30% of labeled images were added to the test set together with 600 slices without lesions of 10 nonpatients. Table 1 displays the makeup of the training set and test set for the GIST detection experiment.

Experimental settings in the fully supervised method (Improved Faster R-CNN) were the following: The operating system was an Ubuntu 18.04, and the hardware environment was an Intel(R) Core(TM) i9-10980XE CPU@3.00 GHz and two TITAN RTX 24 G graphics cards. The programming language used was python3.7, and the framework was the PyTorch-based mmdetection [32]. The backbone of the network was ResNet50, the number of training epochs was 24, the batch size was 8, the optimizer was stochastic gradient descent (SGD), the momentum was 0.9, and the initial learning rate was 0.01, which decays at epochs 16 and 22.

**Table 1.** Composition of the training set and test set.

| Dataset | Object | Number of Slices |
|---|---|---|
| Training set | Small-scale | 413 |
| | Medium-scale | 1412 |
| | Large-scale | 789 |
| Test set | Small-scale | 113 |
| | Medium-scale | 800 |
| | Large-scale | 208 |
| | Nonlesion | 600 |

Datasets used for the semi-supervised method (Improved Faster R-CNN) were the following: The semi-supervised method experiment requires a greater number of unlabeled samples, so the datasets used for the fully supervised method were divided, using 3%, 5%, 10%, and 20% of the data as labeled samples, and the remaining data were unlabeled after the labels were removed to form an unlabeled dataset together with the unlabeled data of 54 patients. The specific division of the train set used for the semi-supervised method is shown in Table 2. The test set was consistent with that used in the fully supervised method experiment.

**Table 2.** Composition of the training set.

| Proportion of Retained Labels | Labeled Data | Unlabeled Data |
|---|---|---|
| 3% | 90 | 6180 |
| 5% | 150 | 6120 |
| 10% | 300 | 5970 |
| 20% | 600 | 5670 |

The experimental settings in the semi-supervised method were as follows: The hardware environment was identical to that in the fully supervised method experiment, and the Improved Faster R-CNN was used as the object detector for the experiments. The batch size was set to 8, the optimizer was stochastic gradient descent (SGD), the initial learning rate was 0.01, the momentum was 0.9, the rounds of iterations were 5, and the times of data augmentation were 4. When training the network with only labeled data, the number of epochs was set to 30 to obtain a better-performing initial model and higher-quality pseudo-labels, with the learning rate decaying at epochs 18 and 26. For better comparison with the experimental results of the fully supervised method, the experimental settings when unlabeled data were used for training were the same as those in the fully supervised learning experiments. The number of epochs was set to 24, and the learning rate decayed at epochs 16 and 22.

The variables used in the self-training method experiments included Q and the threshold. Q indicates the total number of training rounds after adding pseudo-labels, and a higher value of Q indicates a higher proportion of pseudo-labels in the dataset used for training. The pseudo-label selection threshold is one of the criteria used to select pseudo-labels during the training process. If the score of the prediction result is lower than the threshold, it cannot be used as a pseudo-label. The threshold value reflects the correctness of the pseudo-label, and the larger the threshold value is set, the higher the correctness of the pseudo-label.

### *4.2. Ablation Study on Improved Faster R-CNN*

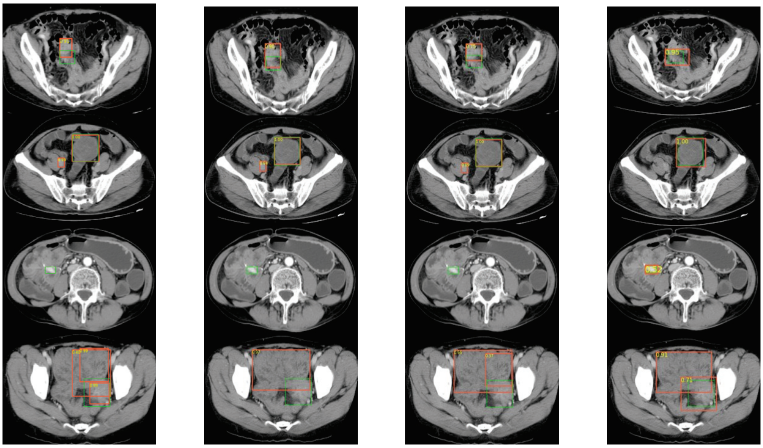
In this section, we describe the designed experiments conducted to demonstrate the effectiveness and superiority of the multiscale module and FEM introduced in the Improved Faster R-CNN. Table 3 lists the experimental results.

**Table 3.** Investigation of different modules introduced to the Faster R-CNN.

| Model | FPN | PAFPN | Ours_FPN | FEM | $AP_s$ | $AP_m$ | $AP_l$ | AP |
|---|---|---|---|---|---|---|---|---|
| Faster R-CNN | | | | | 0.329 | 0.677 | 0.796 | 0.662 |
| | ✓ | | | | 0.395 | 0.692 | 0.806 | 0.680 |
| | | ✓ | | | 0.413 | 0.699 | 0.815 | 0.701 |
| | | | ✓ | | 0.431 | 0.702 | 0.827 | 0.735 |
| | | | | ✓ | 0.342 | 0.671 | 0.799 | 0.667 |
| | ✓ | | | ✓ | 0.401 | 0.695 | 0.811 | 0.682 |
| | | ✓ | | ✓ | 0.415 | **0.711** | 0.814 | 0.705 |
| | | | ✓ | ✓ | **0.435** | 0.704 | **0.829** | **0.739** |

The bold indicates the best result.

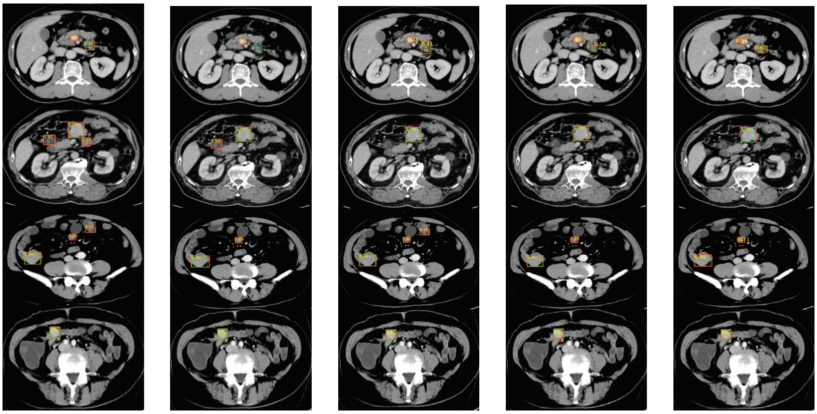
We respectively compare the effects of the Faster R-CNN without the multiscale module, with the initial FPN [27] module, with the PAFPN [33] module, and with the FPN module proposed in this paper, and the experimental results are shown in Table 3 and Figure 4. The data in the table show that the proposed FPN structure increases the AP of the entire item by 0.073, and the AP at all scales is improved by the improved FPN structure, with the most significant improvement for small objects. The suggested multiscale approach surpasses other methods in the AP of both the overall objects and each scale object, proving that the improved FPN is more effective for the task at hand.



(a) base (b) FPN (c) PAFPN (d) Ours_FPN

**Figure 4.** Visual comparison of different feature pyramid networks (the orange bounding box in the figure represents the predicted box, and the green bounding box represents the actual labeled box).

As shown in Table 3 and Figure 5, we compared the test results using only the FEM, using the FPN module together with the FEM, using the PAFPN module together with the FEM, and using our FPN together with the FEM to verify the effectiveness of the FEM. As evident from the results in the table, the AP is slightly improved when using only the FEM without the multiscale module. Using FPN as the multiscale module, the FEM yields an AP improvement of 0.002, using PAFPN 0.004, and using the proposed multiscale module 0.04. Overall, the AP goes from 0.662 to 0.739 with the addition of the multiscale module and the FEM. Although the effect of the FEM on this task is not as significant as that of the multiscale module, a series of comparative experiments also proved that this module can improve the performance of the network.



(a) base (b) FEM (c) FPN+FEM (d) PAFPN+FEM (e) Our_FPN+FEM

**Figure 5.** Visual comparison of the FEM combined with the different feature pyramid networks (the orange bounding box in the figure represents the predicted box, and the green bounding box represents the actual labeled box).

### *4.3. Ablatation Study on the Semi-Supervised Method*

In this section, we detail the ablation experiments of the SSL method to demonstrate that the strategies we introduced to the self-training method are effective in improving the detection results related to the task defined in this paper. In addition, we explored the impact of different settings on the model performance.

To verify the effectiveness of the self-training method used in this paper, we first iteratively optimized the initial model using the original self-training method. We usd the control variable method to test the two initialization hyperparameters, the number of training rounds, and the threshold for pseudo-label selection. The experimental results are shown in Table 4, where Q represents the total number of iterations. When Q = 0, only the labeled data are used for training, and when $Q > 0$, pseudo-labels are gradually added to the train set.

Table 4 shows that as Q rises, the detection accuracy declines. This phenomenon indicates that in the GIST detection task, accuracy improvement cannot be achieved by using the initial self-training method to create pseudo-labels. With the threshold held constant, we observe that as the round of iteration increases, the accuracy of the training set with 3% labeled data decreases faster than that with 20% labeled data. Because the initial model trained with 3% labeled data is less accurate than that with 20% labeled data, a greater number of false pseudo-labels are generated as the number of rounds increases. In the same training set, we can find that the accuracy decreases faster when the pseudo-label selection threshold is 0.5. Using a lower threshold leads to many bounding boxes in nonlesion regions being added to the pseudo-label set, thus generating many false-positive samples in the subsequent round. When the pseudo-label selection threshold is 0.9, the accuracy of the model improves briefly, but as the number of iteration rounds increases, the model tends to overfit the high-confidence data, resulting in a decrease in accuracy. Therefore, if the accuracy of the initial model is too low or the threshold value is not appropriate, there will be too many noisy labels. This result shows that using the original self-training strategy reduces, rather than improves, the detection accuracy due to noisy pseudo labels.

**Table 4.** Investigation of the proportions of labelled data, the thresholds, and the rounds.

| Training Set | Threshold | Q = 0 | Q = 1 | Q = 2 | Q = 3 |
|---|---|---|---|---|---|
| 3% Labeled data | 0.5 | **0.200** | 0.154 | 0.126 | 0.115 |
| | 0.7 | **0.200** | 0.163 | 0.137 | 0.125 |
| | 0.9 | 0.200 | 0.202 | **0.205** | 0.203 |
| 5% Labeled data | 0.5 | **0.312** | 0.281 | 0.241 | 0.239 |
| | 0.7 | **0.312** | 0.288 | 0.246 | 0.244 |
| | 0.9 | 0.312 | **0.314** | 0.313 | 0.312 |
| 10% Labeled data | 0.5 | **0.501** | 0.481 | 0.472 | 0.469 |
| | 0.7 | **0.501** | 0.482 | 0.475 | 0.471 |
| | 0.9 | 0.501 | 0.503 | **0.505** | 0.504 |
| 20% Labeled data | 0.5 | **0.637** | 0.615 | 0.608 | 0.604 |
| | 0.7 | **0.637** | 0.621 | 0.614 | 0.611 |
| | 0.9 | 0.637 | **0.638** | 0.636 | 0.636 |

The bold indicates the best result.

Table 5 shows the the experimental findings on the 10% annotated dataset without data augmentation, with data augmentation using horizontal flip, vertical flip, random rotation, random noise, and affine transformation to confirm the impact of various data augmentation approaches. The use of data augmentation techniques other than random noise can increase the accuracy of detection. One of the four data augmentation techniques (horizontal flip, vertical flip, rotation, and affine transformation) was chosen at random in the experiment. There is some variation in the training outcomes because the data augmentation method was chosen at random. As a result, the findings of the subsequent experiments involving data augmentation were averaged after four rounds of training.

**Table 5.** Influence of using different data augmentation methods on the 10% labeled dataset.

| Data Augmentation | $AP_{0.5}$ | $AP_{0.75}$ | AP |
|---|---|---|---|
| Base | 0.793 | 0.580 | 0.501 |
| Horizontal flip | 0.801 | 0.585 | 0.506 |
| Verical flip | 0.795 | 0.578 | 0.504 |
| Random rotation | 0.794 | 0.581 | 0.503 |
| Affine transformation | 0.794 | 0.582 | 0.502 |
| Random noise | 0.801 | 0.576 | 0.491 |

The underline indicates the worse results after using data augmentation.

To demonstrate the effectiveness of the dynamic threshold, the IOU constraint, and the Mixup used in the self-training method, we present ablation studies on the 10% labeled data. Table 6 shows that after applying the dynamic threshold and the IOU constraint, the results are improved with iteration. In contrast, as the training iterates, the accuracy of the model using the original self-training method declines. We can also observe that employing Mixup results in a modest rise in AP, proving that Mixup enhances the network's generalization ability and improves the model's performance.

**Table 6.** Investigation of using different strategies in the self-training method.

| Model | Dynamic Threshold | IOU Constraint | Mixup | Q = 0 | Q = 3 | Q = 5 |
|---|---|---|---|---|---|---|
| Improved Faster R-CNN | ✓ | | | 0.501 | 0.515 | 0.518 |
| | | ✓ | | 0.501 | 0.517 | 0.518 |
| | ✓ | ✓ | | 0.501 | 0.519 | 0.521 |
| | ✓ | ✓ | ✓ | 0.501 | **0.521** | **0.524** |

The bold indicates the best result.

### *4.4. Comparison Experiments and Analysis*

In this section, we describe the series of comparative experiments conducted to prove the superiority of the Improved Faster R-CNN and the proposed self-training method.

As shown in Table 7 and Figure 6, the Improved Faster R-CNN outperformed the other mainstream object detection algorithms, including the one-stage object detector, two-stage object detector, and anchor-free object detector, which demonstrates that our network has superiority by virtue of the multiscale module and the FEM.

**Table 7.** Comparison with the state-of-the-art object detection algorithms.

| Model | $AP_{0.5}$ | $AP_{0.75}$ | AP |
|---|---|---|---|
| Faster R-CNN [1] | 0.923 | 0.771 | 0.662 |
| Mask R-CNN [34] | 0.933 | 0.802 | 0.702 |
| Cascade R-CNN [2] | 0.914 | 0.798 | 0.703 |
| YOLOv3 [3] | 0.933 | 0.474 | 0.484 |
| RetinaNet [4] | 0.922 | 0.783 | 0.673 |
| FCOS [35] | 0.860 | 0.543 | 0.516 |
| Improved Faster R-CNN | **0.961** | **0.802** | **0.739** |

The bold indicates the best result.

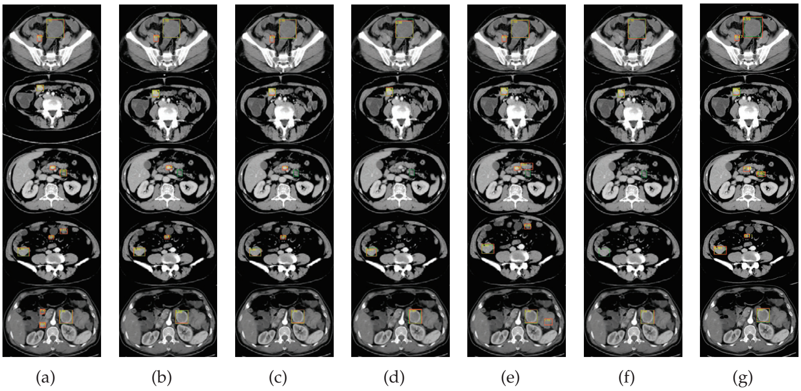


**Figure 6.** Visual comparison of the different object detection algorithms including (**a**) Faster R-CNN, (**b**) Mask R-CNN, (**c**) Cascade R-CNN, (**d**) YOLOv3, (**e**) RetinaNet, (**f**) FCOS, and (**g**) Improved Faster R-CNN. The orange bounding box in the figure represents the predicted box, and the green bounding box represents the actual labeled box.

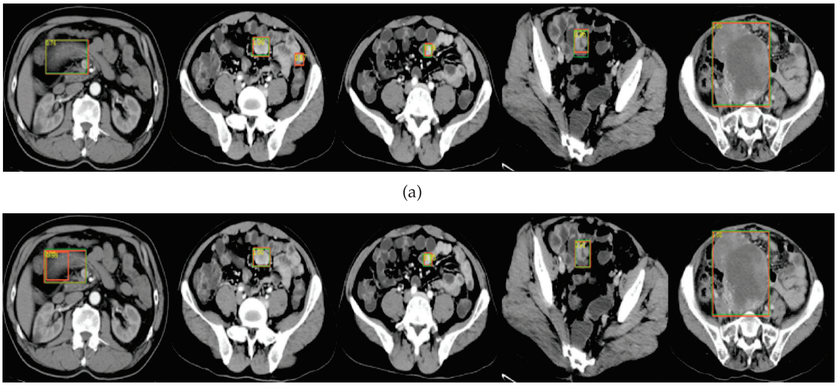
In Table 8, we present the improvement in accuracy when using our semi-supervised method compared to CSD [23], STAC [24], and Instant-teaching [36] on all datasets. It demonstrates that the network has higher robustness for all datasets since the dynamic threshold and the IOU constraint guarantee accurate pseudo-labels.

**Table 8.** Comparison with state-of-the-art semi-supervised methods.

| Dataset | Method | $AP_{0.5}$ | $AP_{0.75}$ | AP |
|---|---|---|---|---|
| 3% labeled data | CSD | 0.283 | 0.227 | 0.198 |
| | STAC | 0.291 | 0.231 | 0.191 |
| | Instant-teaching | 0.288 | 0.246 | 0.197 |
| | Ours | **0.301** | **0.257** | **0.204** |
| 5% labeled data | CSD | 0.574 | 0.346 | 0.351 |
| | STAC | **0.597** | 0.319 | 0.348 |
| | Instant-teaching | 0.574 | 0.300 | 0.341 |
| | Ours | 0.587 | **0.412** | **0.355** |
| 10% labeled data | CSD | 0.715 | 0.511 | 0.499 |
| | STAC | 0.759 | 0.587 | 0.508 |
| | Instant-teaching | 0.781 | 0.604 | 0.504 |
| | Ours | **0.835** | **0.611** | **0.524** |
| 20% labeled data | CSD | 0.891 | 0.721 | 0.639 |
| | STAC | 0.901 | 0.701 | 0.638 |
| | Instant-teaching | 0.913 | 0.728 | 0.648 |
| | Ours | **0.932** | **0.731** | **0.659** |

The bold indicates the best result.

Finally, we compared the semi-supervised method with the fully supervised Improved Faster R-CNN on the fully labeled dataset, using all the labeled data for initial training and adding the unlabeled CT images of only 54 patients for the pseudo-label generation. The experimental results are shown in Table 9 and Figure 7. The comparison reveals that the semi-supervised method can use the information in the unlabeled data to produce detection results that are marginally better than those of the fully supervised method after using unlabeled data. It proves that the addition of unlabeled data has no impact on the model's performance. However, the improvement in the model performance is not substantial because the unlabeled data used in this comparison experiment were insufficient and only contained images of 54 patients.


(a)

(b)

**Figure 7.** Visual comparison of (**a**) the fully supervised method and (**b**) semi-supervised method. The orange bounding box in the figure represents the predicted box, and the green bounding box represents the actual labeled box.

**Table 9.** Comparison with the fully supervised Improved Faster R-CNN.

| Method | $AP_{0.5}$ | $AP_{0.75}$ | AP |
|---|---|---|---|
| Fully supervised Improved Faster R-CNN | 0.959 | **0.799** | 0.739 |
| Semi-supervised Improved Faster R-CNN | **0.960** | 0.797 | **0.740** |

The bold indicates the best result.

## 5. Conclusions

To address the problems of large object scale variation, confusing background, and the challenge of obtaining labeled data in the detection of GIST, we propose a semi-supervised object detection method using self-training in this study. The method uses only a small amount of labeled data supplemented by a sizable amount of unlabeled data and fully exploits the information contained in the unlabeled data. Through comparison with existing methods and ablation studies of each module, the feasibility of the proposed method was proven, and the detection accuracy of the model increased without extra labeling costs.

Although the improved scheme for GIST detection in this paper has achieved good results, there are still some limitations that should be noted. In the semi-supervised learning method, the best prediction results are used as pseudo-labels for subsequent training. The challenging samples—those that the model has not yet learned well—are not used effectively. Therefore, we can try to combine semi-supervised learning and active learning to find challenging samples by active learning and then manually label the challenging samples for training.

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

**Funding:** This work was partially supported by the National Natural Science Foundation of China, Nos. 61973250, 62073218, 61973249, 61902316, 61902313, 62002271, 82150301, 62133012, 62273232, 62273231. Young science and technology nova of Shaanxi Province (2022KJXX-73) and the Fundamental Research Funds for the Central Universities under grant No. XJS210310. The Shaanxi Provincial Department of Education Serving Local Scientific Research (19JC038), the Key Research and Development Program of Shaanxi (2021GY-077), the Young Science and Technology Nova of Shaanxi Province (2022KJXX-73), the Shanghai Municipal Health Commission (202140512), and the Shanghai Stroke Association (SSA-2020-023). National Key R and D program of China (2022YFB4300700), the Key R and D programs of Shaanxi Province (2021ZDLGY02-06), Qin Chuangyuan project (2021QCYRC4-49), Qinchuangyuan Scientist+Engineer (2022KXJ-169), National Defense Science and Technology Key Laboratory Fund Project (6142101210202).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

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

## References

1. 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**, *39, 91–99.* [CrossRef] [PubMed]
2. 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, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
3. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
4. 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.

5. Nguyen, D.T.; Mummadi, C.K.; Ngo, T.P.N.; Nguyen, T.H.P.; Beggel, L.; Brox, T. Self: Learning to filter noisy labels with self-ensembling. *arXiv* **2019**, arXiv:1910.01842.
6. Park, S.; Han, S.; Kim, S.; Kim, D.; Park, S.; Hong, S.; Cha, M. Improving unsupervised image clustering with robust learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 12278–12287.
7. Qiao, S.; Shen, W.; Zhang, Z.; Wang, B.; Yuille, A. Deep co-training for semi-supervised image recognition. In Proceedings of the European Conference on Computer Vision (Eccv), Munich, Germany, 8–14 September 2018; pp. 135–152.
8. Li, J.; Socher, R.; Hoi, S.C. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv* **2020**, arXiv:2002.07394.
9. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
10. Cireşan, D.C.; Giusti, A.; Gambardella, L.M.; Schmidhuber, J. Mitosis detection in breast cancer histology images with deep neural networks. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Nagoya, Japan, 22–26 September 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 411–418.
11. Setio, A.A.A.; Ciompi, F.; Litjens, G.; Gerke, P.; Jacobs, C.; Van Riel, S.J.; Wille, M.M.W.; Naqibullah, M.; Sánchez, C.I.; Van Ginneken, B. Pulmonary nodule detection in CT images: False positive reduction using multi-view convolutional networks. *IEEE Trans. Med. Imaging* **2016**, *35*, 1160–1169. [CrossRef] [PubMed]
12. Rajpurkar, P.; Irvin, J.; Zhu, K.; Yang, B.; Mehta, H.; Duan, T.; Ding, D.; Bagul, A.; Langlotz, C.; Shpanskaya, K.; et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv* **2017**, arXiv:1711.05225.
13. Sedik, A.; Hammad, M.; El-Samie, A.; Fathi, E.; Gupta, B.B.; El-Latif, A.; Ahmed, A. Efficient deep learning approach for augmented detection of Coronavirus disease. *Neural Comput. Appl.* **2022**, *34*, 11423–11440. [CrossRef] [PubMed]
14. Xie, F.; Zhou, Y.; Guan, Z.; Duan, Q. Joint multi-model detection of small intestinal mesenchymal tumors based on deformable convolution. In *Journal of Northwest University (Natural Science Edition)*; Xi Bei Da Xue Xue Bao Bian Ji Bu: Xi'an, China, 2021. (In Chinese)
15. Chang, X.; Ren, P.; Xu, P.; Li, Z.; Chen, X.; Hauptmann, A. A comprehensive survey of scene graphs: Generation and application. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *45*, 1–26. [CrossRef] [PubMed]
16. Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C.A. Mixmatch: A holistic approach to semi-supervised learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5050–5060.
17. Miyato, T.; Maeda, S.I.; Koyama, M.; Ishii, S. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1979–1993. [CrossRef] [PubMed]
18. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 596–608.
19. Lee, D.H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Proceedings of the Workshop on Challenges in Representation Learning, Atlanta, GA, USA, 21 June 2013; Volume 3, p. 896.
20. Iscen, A.; Tolias, G.; Avrithis, Y.; Chum, O. Label propagation for deep semi-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5070–5079.
21. Xie, Q.; Luong, M.T.; Hovy, E.; Le, Q.V. Self-training with noisy student improves imagenet classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10687–10698.
22. Mukherjee, S.; Awadallah, A.H. Uncertainty-aware self-training for text classification with few labels. *arXiv* **2020**, arXiv:2006.15315.
23. Jeong, J.; Lee, S.; Kim, J.; Kwak, N. Consistency-based semi-supervised learning for object detection. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 10759–10768.
24. Sohn, K.; Zhang, Z.; Li, C.L.; Zhang, H.; Lee, C.Y.; Pfister, T. A simple semi-supervised learning framework for object detection. *arXiv* **2020**, arXiv:2005.04757.
25. Yang, Q.; Wei, X.; Wang, B.; Hua, X.S.; Zhang, L. Interactive self-training with mean teachers for semi-supervised object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 5941–5950.
26. Zhang, F.; Pan, T.; Wang, B. Semi-supervised object detection with adaptive class-rebalancing self-training. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2022; Volume 36, pp. 3252–3261.
27. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
28. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
29. 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.
30. Park, J.; Woo, S.; Lee, J.Y.; Kweon, I.S. Bam: Bottleneck attention module. *arXiv* **2018**, arXiv:1807.06514.
31. Radosavovic, I.; Dollár, P.; Girshick, R.; Gkioxari, G.; He, K. Data distillation: Towards omni-supervised learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4119–4128.
32. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv* **2019**, arXiv:1906.07155.

33. 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, 18–23 June 2018; pp. 8759–8768.
34. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
35. 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, 27–28 October 2019; pp. 9627–9636.
36. Zhou, Q.; Yu, C.; Wang, Z.; Qian, Q.; Li, H. Instant-teaching: An end-to-end semi-supervised object detection framework. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 4081–4090.

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

*electronics*

*Article*

# Learning Data-Driven Propagation Mechanism for Graph Neural Network

**Yue Wu [1], Xidao Hu [1], Xiaolong Fan [2], Wenping Ma [3,*] and Qiuyue Gao [1]**

[1] School of Computer Science and Technology, Xidian University, Xi'an 710071, China
[2] School of Electronic Engineering, Xidian University, Xi'an 710071, China
[3] School of Artificial Intelligence, Xidian University, Xi'an 710071, China
* Correspondence: wpma@mail.xidian.edu.cn

**Abstract:** A graph is a relational data structure suitable for representing non-Euclidean structured data. In recent years, graph neural networks (GNN) and their subsequent variants, which utilize deep neural networks to complete graph analysis and representation, have shown excellent performance in various application fields. However, the propagation mechanism of existing methods relies on hand-designed GNN layer connection architecture, which is prone to information redundancy and over-smoothing problems. To alleviate this problem, we propose a data-driven propagation mechanism to adaptively propagate information between layers. Specifically, we construct a bi-level optimization objective and use the gradient descent algorithm to learn the forward propagation architecture, which improves the efficiency of learning different layer combinations in multilayer networks. The experimental results of the model on seven benchmark datasets demonstrate the effectiveness of the proposed method. Furthermore, combining this data-driven propagation mechanism with models, such as Graph Attention Networks, can consistently improve the performance of these models.

**Keywords:** graph neural network; propagation mechanism; data-driven method; deep learning

## 1. Introduction

Graphs are data structures that model a set of objects (nodes) and their relationships (edges). Graphs can be irregular and have variable-sized unordered nodes, and nodes can have different numbers of neighbors. As a consequence, while some important operations (e.g., convolutions [1]) can be easily applied to the image domain [2], it is difficult to apply to the graph domain. In addition, a key assumption of existing deep learning algorithms is that the data samples are independent of each other. For graphs, however, there are edges between each data sample (node) and other data samples (nodes) that capture the interdependencies between instances. Due to the powerful representational power of graph structures, the study of graph analysis using machine learning methods has received increasing attention. Researchers have defined and designed a neural network architecture for processing graph data. This structure has become a new research hotspot—"graph neural network (GNN)", which achieves excellent performance and interpretability on graph-structured data.

For example, papers in a citation network are linked to each other by citations, and GNNs can classify each paper into a different group [3–6]. In the fields of chemistry and medicine, molecules can be modeled as graphs, and their biological activities can be identified by GNNs for drug development [7–10]. In the field of computer vision, GNNs can identify objects depicted by 3D point clouds and explore their topology [11–15]. In the traffic system, GNNs can accurately predict the traffic speed and traffic flow in the traffic network for route planning and flow control [16,17].

GNNs are used to learn node representations (node embeddings), which can simultaneously model node features and graph topology information. In addition, GNNs utilize the relationships (edges) between nodes of a graph to propagate information rather than

**Citation:** Wu, Y.; Hu, X.; Fan, X.; Ma, W.; Gao, Q. Learning Data-Driven Propagation Mechanism for Graph Neural Network. *Electronics* **2023**, *12*, 46. https://doi.org/10.3390/electronics12010046

Academic Editor: Stefanos Kollias

Received: 21 November 2022
Revised: 12 December 2022
Accepted: 20 December 2022
Published: 22 December 2022

treating them as features of nodes. Among them, models such as Graph Convolutional Networks (GCN) [3] and Graph Attention Networks (GAT) [18] follow a neighborhood aggregation (message passing) scheme. These models learn to iteratively aggregate the hidden features of each node in the graph and its neighbors as its new hidden features, where the iterations are parameterized by neural network layers. Theoretically, the aggregation process of L iterations fuses the structural information of each node at each layer, which can simultaneously learn the topology and the distribution of node features in the neighborhood.

However, in practice, a deeper version of the model with more information is likely to perform worse. For example, the best performance of GCN and GAT experiments on the Planetoid dataset [19] is achieved with a two-layer model, and increasing the number of layers will reduce the performance. A similar degradation of learning for computer vision problems is addressed by residual connections [20], which greatly aids the training of deep models. However, even with residual connections, GCNs with more layers do not perform as well as two-layer GCNs on datasets such as the Citation Network datasets PubMed [21], CiteSeer, and Cora [22].

We believe that the structure of different nodes and their neighborhoods (subgraphs) in the graph has a great influence on the result of neighborhood aggregation. The rate of expansion, or the growth rate of the radius of influence, is characterized by the mixing time of random walks and varies significantly across subgraphs of different structures. Therefore, the same number of iterations can result in very different local distributions. For example, consider a node at the center of the graph and a node at the edge of the graph to start an expansion of a random walk. After the same number of iterative layers, the nodes that may be located in the center of the graph already contain basically all the information of the graph, so only a small amount of information from other layers needs to be aggregated. At this time, if all the information of each layer is aggregated, it will cause redundancy. The nodes located at the edge of the graph may contain only a small amount of information, and more information needs to be aggregated to perceive the structure of the graph.

To adaptively adjust the influence radius of each node and task, we propose a data-driven propagation mechanism that learns to selectively acquire information from various layers. Finally, each node can selectively obtain low-order local structural information and high-order neighborhood information, thereby effectively avoiding the problems of local structural information degradation and information redundancy and enhancing the representation ability of GNNs. Additionally, stacking too many layers and non-linear transformations can lead to over-smoothing issues, where node representations tend to converge to a fixed value, resulting in degraded model performance. To alleviate this problem, we add an identity map to the convolution operation to improve the network performance.

Since learning a combination of different layers in a multilayer network is computationally expensive, we adopt a differentiable approach to reduce the computational cost. The model achieves good results on the node classification task, demonstrating the effectiveness of the proposed data-driven propagation mechanism. In conclusion, we outline the main contributions of this paper as follows:

(1) We propose a data-driven propagation mechanism (GraphSAP), which adaptively learns the connections between different layers, enabling nodes to selectively fuse low-order local structural information while acquiring high-order neighborhood information.

(2) We add the identity map to the neighbor aggregation function of the GraphSAP model and use a differentiable algorithm during training to make the model more efficient while maintaining high performance.

(3) We provide a quantitative comparison of the node classification task under different datasets, showing the good performance of the model.

## 2. Related Work

### 2.1. Graph Neural Networks

The concept of graph neural networks was first proposed in [23] and further clarified by Scarselli et al. [24], and many variants [18,25] have been proposed over the past few years. Ref. [24] is the first paper to propose a graph neural network model, which applies neural networks to graph-structured data, and elaborates the structure, calculation method, optimization algorithm, and implementation of the neural network model in detail.

GNN is a new research hotspot that emerged after the maturity of convolutional neural networks (CNN) [1] to process non-Euclidean data. Some existing studies try to apply the methods used by CNN to GNN to utilize the excellent abilities of CNN. The existing deep GNN model adds other operations to the convolution operation to alleviate over-smoothing or aggregates different layers. Among the contributions of stacking more layers of CNNs, ResNet [20] and DenseNet [26] are excellent methods that can be seen in many deep networks today. JKNet [27] is inspired by ResNet, but it does not achieve good performance by stacking multiple layers like ResNet and can not fully achieve the representation ability of GNN. These methods are all hand-crafted networks. Therefore, we cannot directly apply the method of CNN to GNN, but needs to convert these methods to make GNN obtain better performance. The focus of our work is to better exploit the representational power of GNNs.

### 2.2. Data-Driven Methods

Hand-designed interlayer connection network structures have achieved great success in the past. The emergence of ResNet [20] and DenseNet [26] showed the importance of residual and dense connections for the design of deep networks and had a huge impact on the design of deep neural networks. With the continuous development of deep neural networks and the continuous invention and utilization of various models and new modules, people gradually realize that developing a new neural network structure is more time-consuming and labor-intensive.

People have begun to explore how to use existing machine learning knowledge to independently build networks suitable for business scenarios. Automated Machine Learning (AutoML) is one of the hottest fields in machine learning and deep learning in recent years. Several recent works have demonstrated the feasibility of automated learning [28] and designed some models that go beyond hand-designed ones, such as [29,30]. Using the dataset as the basis for training the network, various network structures can be designed. For example, if you have a four-layer network, then mathematically, there are 15 combinations of layer-to-layer connections in total. Ideally, given sufficient resources and time, data-driven learning methods can simulate all connections between layers, which would cover all hand-designed network structures. A representative method is the Neural Architecture Search (NAS) algorithm, such as [31]. In NAS, the network architecture is mainly designed from three parts: search space, search strategy, and evaluation strategy. The data-driven approach is also a method in the field of AutoML, which adaptively learns a network model suitable for the data based on the existing data, which is used in our work.

## 3. Background

Given an undirected graph $\mathcal{G} = (V, E)$ with node features $X \in \mathbb{R}^{n \times d_i}$, where $V$ and $E$ denote node and edge sets, respectively. $n$ represents the number of nodes, and $d_i$ is the dimension of node features. We use $N(v)$ to represent the first-order neighbors of a node $v$ in $\mathcal{G}$, i.e., $N(v) = \{u \in V | (v, u) \in E\}$. In addition, we use the set $\tilde{N}(v)$ to denote the set of neighbors, including oneself, i.e., $\tilde{N}(v) = \{v\} \cup \{u \in V | (v, u) \in E\}$. Let $\tilde{\mathcal{G}}$ be the graph obtained by adding a self-loop to every $v \in V$. The hidden feature of node $v$ learned by the $l$-th layer of the model is denoted by $h_v^{(l)} \in \mathbb{R}^{d_h}$, where $d_h$ denotes the dimension of the hidden features. For simplicity of illustration, we assume that it is the same between layers. Let $A$ denote the adjacency matrix and $D$ the diagonal degree matrix. Consequently, the adjacency matrix and diagonal degree matrix of $\tilde{G}$ is defined to be

$\tilde{A} = A + I$ and $\tilde{D} = D + I$, respectively. The normalized graph Laplacian matrix is defined as $L = I_n - D^{-1/2}AD^{-1/2}$, which is a symmetric positive semidefinite matrix.

### *3.1. Graph Convolutional Network*

Kipf et al. [3] proposed the Graph Convolutional Network (GCN) model, which can be described as the "pioneering work" of GNN. GCN uses approximation techniques to derive a simple and efficient model that enables convolution operations in image processing to be easily used for graph-structured data processing. Inspired by GCNs, various new graph neural networks are emerging. The form of GCN can be expressed as:

$$h_i^{(l+1)} = \sigma\left(b^{(l)} + \sum_{j \in N(i)} \frac{1}{c_{ij}} h_j^{(l)} W^{(l)}\right) \tag{1}$$

where $c_{(ij)} = \sqrt{|N(i)|}\sqrt{|N(j)|}$ is a regularization term, $W^{(l)}$ and $b^{(l)}$ are trainable parameters, and $\sigma$ is a non-linear activation function, e.g., a ReLU.

In principle, deeper versions of GCN models that can capture more information will perform better. We conduct node classification experiments on the Cora dataset using GCNs with 2-layer, 4-layer, 6-layer, and 16-layer network structures, respectively, to analyze the performance of GCNs with different layers. The experimental result is shown in Figure 1. The best performance of GCN on the node classification task on the Cora dataset is achieved with a 2-layer model, and increasing the number of layers will reduce the performance. This is due to the over-smoothing problem; as the number of layers in the network increases and the number of iterations increases, the hidden layer representation of each node tends to converge to the same value.

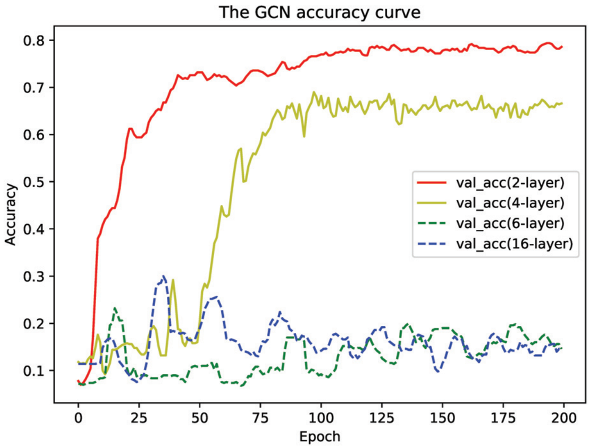


**Figure 1.** Performance of GCNs with different numbers of layers on the node classification task on the Cora dataset.

### *3.2. Deep GNNs*

In order to better exploit information from neighborhoods of differing localities and improve the over-smoothing problem of deep GNN models, models such as Jumping Knowledge Networks [27] and GCNII [32] proposed a network structure similar to ResNet [20] structure. These models are roughly represented as follows:

$$h_v^{(l+1)} = \sigma(W^{(l+1)} \cdot aggregate(\{h_u^{(l)}, u \in \tilde{N}(v)\})) \tag{2}$$

$$h_v^{(final)} = layer\_aggregation(h_v^{(1)}, h_v^{(2)}, \ldots, h_v^{(n)}) \tag{3}$$

where $aggregate$ represents aggregation operations between nodes and $layer\_aggregation$ is the layer aggregation function, indicating that all representations of the middle layer are aggregated in the last layer. However, this hand-designed way of aggregating the features of all layers may result in information redundancy.

Many GNN models [33–35] obtain node features via a message-passing pattern [7,36,37], where the representation of each node is learned by iteratively aggregating the embeddings ("messages") of its neighbors. APGCN [33] sets each node as an extra unit when the message is passed, which outputs a value that controls whether the communication should continue. This method can better control the information propagation of nodes to combine information from more distant neighbors, but it cannot aggregate information from different layers. To address the above issues, we use GCN as a benchmark and design an adaptive learning method for inter-layer aggregation. Compared with hand-designed networks, it can automatically learn the network aggregation architecture to fully exploit the representational capabilities of GNNs.

## 4. GraphSAP Network

### 4.1. Model Analysis

To improve the representation ability of the network model, we design a data-driven propagation mechanism that adaptively learns the connections of different layers, that is, the aggregation of different neighbors. We use GCN as the baseline network in our network and alleviate the over-smoothing problem in deep networks by adding an identity map to the convolution operation. Formally, we define the $l$-th layer of our GraphSAP as:

$$H^{(l+1)} = \sigma\left(\kappa_l \tilde{L} H^l \left((1-\beta_l)I_n + \beta_l W^l\right)\right) \tag{4}$$

where $\kappa_l$ and $\beta_l$ are two hyperparameters, and $\tilde{L} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ is the graph convolution matrix with the renormalization trick. Compared to the vanilla GCN model (Equation (1)), we add the identity map $I_n$ to the $l$-th weight matrix $W^l$.

Each intermediate layer is computed from all its predecessors:

$$layer^{(j)} = \sum_{i<j} o^{(i,j)} layer^{(i)} \tag{5}$$

where $layer$ can be obtained by Equation (4), and $o^{(i,j)}$ denotes the connection state between $layer^{(i)}$ and $layer^{(j)}$.

The network architecture of our GraphSAP is shown in Figure 2. The main difference between our model and the existing models is that we design an adaptive learning network architecture based on a data-driven propagation mechanism instead of relying on handcrafted designs. We incorporate identity maps into convolutions to guarantee model performance and then use a data-driven adaptive approach to learn the best-performing network aggregation structure. Our proposed network achieves good results in the node classification task, demonstrating the feasibility of our proposed method.

Identity maps play an important role in preventing performance degradation in deep models, so we add identity maps to the model's operations. Generally speaking, identity mapping is to add the identity matrix to the weight matrix, which can alleviate the over-smoothing problem of the model due to the increase in the number of network layers. Frequent interactions between different dimensions of the feature matrix [38] will degrade the performance of the model in semi-supervised tasks, whereas direct mapping of the smooth representation $\tilde{L}H^l$ to the output will reduce this interaction.

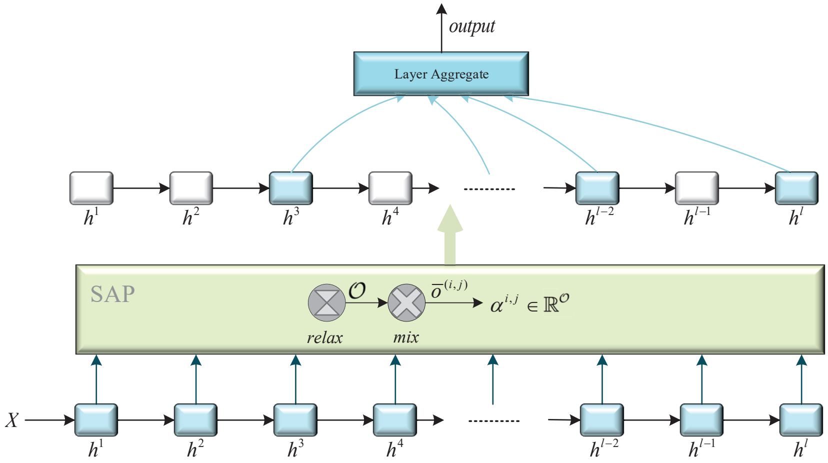


**Figure 2.** Network architecture of GraphSAP. SAP is based on data-driven learning at different layers. Where $l$ is the number of layers of the network, $h^{(l)}$ denotes the hidden features learned by the node at layer $l$, *relax* denotes relaxation operation, *mix* denotes Equation (6), and $X$ denotes the initial node features.

### *4.2. Data-Driven Propagation Mechanism*

In this subsection, we first introduce the proposed propagation mechanism. We first introduce the differences and connections between our data-driven propagation mechanism and existing propagation strategies, such as Learning to Propagation (L2P) [39]. Although both L2P and our proposed GraphSAP belong to adaptive propagation, there are still differences between the two methods. Our GraphSAP learns whether neighbor node features of nodes at different levels are aggregated. L2P considers that different nodes may require different propagation layers, so it needs to learn the order of neighbor nodes. Next, we introduce methods for continuous operations between layers and, finally, optimization methods to speed up the learning time.

JKNet [27] aggregates the features of nodes of all layers to get the final feature representation, as shown in Equation (3). Our method obtains a layer connection operation space and adaptively learns aggregations between different layers, as shown in Equation (5), each directed edge $(i, j)$ is associated with the edge state $o^{(i,j)}$. Our final task is to find a suitable connection method for each layer. The combination of these operations is discontinuous and learning in discrete spaces is very difficult.

To make the search space continuous, we relax the classification selection for a specific operation to a softmax of all possible operations:

$$\bar{o}^{(i,j)}(layer) = \sum_{o \in \mathcal{O}} \frac{exp\left(\alpha_o^{i,j}\right)}{\sum_{o' \in \mathcal{O}} exp\left(\alpha_{o'}^{i,j}\right)} o(layer) \tag{6}$$

where $\mathcal{O}$ is the set of all candidate aggregation operations (e.g., *identity*, *maxpooling*, and *zero*), and each operation represents some function $o(\cdot)$ to be applied to the layer, and the operation mixing weights for a pair of layers $(i, j)$ are parameterized by a vector $\alpha^{i,j}$ of dimension $|\mathcal{O}|$. *layer* represents the GNN layer, as shown in Equation (5). The layer aggregation operation of GraphSAP is shown in Equation (4), and the node features of the last layer can be obtained in the following ways:

$$layer^{(l)} = [o(H^l), ..., o(H^{l-1})] \tag{7}$$

where $o$ is a classification operation, indicating whether this layer participates in information transmission. The final feature of the nodes can be expressed as

$$Z = softmax\left(layer^{(l)}\right) \tag{8}$$

When training the network, we use $L_{train}$ and $L_{val}$ to denote the training and validation losses. Both losses depend not only on the architecture $\alpha$, but also on the weights $w$ in the network. The goal of our method is to find $\alpha^*$ that minimizes $\mathcal{L}_{val}(w^*, \alpha^*)$, where $w^*$ is the weight that minimizes $\mathcal{L}_{train}$. Thus, our model actually needs to solve a bi-level optimization [40] problem:

$$\min_{\alpha \in \mathcal{A}} \quad \mathcal{L}_{val}(w^*(\alpha), \alpha^*) \tag{9}$$

$$s.t. \quad w^*(\alpha) = argmin_w\, \mathcal{L}_{train}(w, \alpha) \tag{10}$$

where $\alpha$ denotes the network architecture, and $\omega^*(\alpha)$ denotes the weight of this architecture after training. In our experiments, we choose to use the cross-entropy loss for our semi-supervised node classification task:

$$\mathcal{L} = -\sum_{i \in \mathcal{Y}_L} Y_i \log Z_i \tag{11}$$

where $\mathcal{Y}_L$ is the set of node indices with labels, $Y_i$ denotes the predicted label of node $i$, and $Z_i$ is the final feature representation of node $i$. This cross-entropy function is used for adaptive generative network structure training and task model training.

We train the network using a one-shot differentiable method; the optimization details are given in Algorithm 1. In addition, we use the gradient-based approximation method [41–43] to update the operation parameter $\alpha$ to save training time, as follows:

$$\bigtriangledown_{\alpha}\, \mathcal{L}_{val}(w^*(\alpha), \alpha) \approx \bigtriangledown_{\alpha} \mathcal{L}_{val}(w - \gamma \bigtriangledown_{w} \mathcal{L}_{train}(w, \alpha), \alpha) \tag{12}$$

where $w$ denotes the current weights maintained by the algorithm, and $\gamma$ is the learning rate for a step of inner optimization. We use only a single training step to adjust $w$ to approximate $w^*(\alpha)$ without fully solving the internal optimization by training until convergence (Equation (10)).

**Algorithm 1:** Data-Driven Propagation Mechanism (SAP).

```
Input: The aggregation operations $\mathcal{A}$, the number of top performance network $k$,
       the epochs $N$ for learning.
Output: The $k$ aggregation operations $\mathcal{A}_k$.
while $t = 1, ..., N$ do
    Compute the validation loss $\mathcal{L}_{val}(w - \gamma\nabla_w\mathcal{L}_{train}, \alpha)$;
    Update network $\alpha$ by descending $\nabla_\alpha\mathcal{L}_{val}$;
    Compute the train loss $\mathcal{L}_{train}(w, \alpha)$;
    Update weights $w$ by descending $\nabla_w\mathcal{L}_{train}$ with the network $\alpha$;
Derive the final network structure based on the learned $\alpha$.
Return
```

After training, we take the top-$k$ operations with good performance in each layer (in our experiments, we set $k = 1$), such as the maximum weight in Equation (6), to form our model. After adaptive learning is complete, we train from scratch using the best-performing model and adjust it based on the validation data to receive the final parameters.

## 5. Experiment

### *5.1. Datasets*

To verify the effectiveness of our proposed algorithm, we use seven benchmark datasets to perform the node classification task. Table 1 summarizes the statistics of the dataset. We conduct experiments on three citation network datasets: PubMed [21], CiteSeer, and Cora [22]. Each of their nodes represents a paper, and each edge represents a citation relationship between two papers. The dataset contains bag-of-words features for each paper (node). The task is to classify papers into different topics according to a citation network, i.e., node classification. We also introduce four new datasets for the node classification task: Coauthor CS, Coauthor Physics, Amazon Computers, and Amazon Photo [44]. Descriptions of these new datasets are mentioned below. We split the nodes in all graphs into 60%, 20%, and 20%for training, validation, and testing.

**Table 1.** Dataset statistics.

| Datasets | Nodes | Edges | Features | Classes |
|---|---|---|---|---|
| Cora | 2708 | 5278 | 1433 | 7 |
| CiteSeer | 3327 | 4552 | 3703 | 6 |
| PubMed | 19,717 | 44,324 | 500 | 3 |
| Coauthor CS | 18,333 | 81,894 | 6805 | 15 |
| Coauthor Physics | 34,493 | 247,962 | 8415 | 5 |
| Amazon Computers | 13,752 | 245,778 | 767 | 10 |
| Amazon Photo | 7487 | 119,043 | 745 | 8 |

**Cora.** The Cora dataset consists of machine learning papers divided into the following seven categories: Case Based; Genetic Algorithms; Neural Networks; Probabilistic Methods; Reinforcement Learning; Rule Learning; Theory.

**CiteSeer.** The Citeseer dataset is a portion of papers selected from the CiteSeer Digital Papers repository and is grouped into the following six categories: Agents; AI; DB; IR; ML; HCI.

**PubMed.** The PubMed dataset includes 19,717 scientific publications on diabetes from the Pubmed database, divided into three categories: Diabetes Mellitus, Experimental; Diabetes Mellitus Type 1; Diabetes Mellitus Type 2.

**Coauthor CS and Coauthor Physics.** They are coauthorship graphs based on the Microsoft Academic Graph from the KDD Cup 2016 challenge. Nodes in the dataset represent authors and are connected by an edge if two authors coauthored a paper. Node features represent keywords of each author's papers, and category labels represent each author's most active research area.

**Amazon Computers and Amazon Photo.** They are fragments of the Amazon co-purchase graph [44], where nodes represent items, edges represent two items that are frequently purchased together, node features are bag-of-words-encoded product reviews, and category labels represent product classifications.

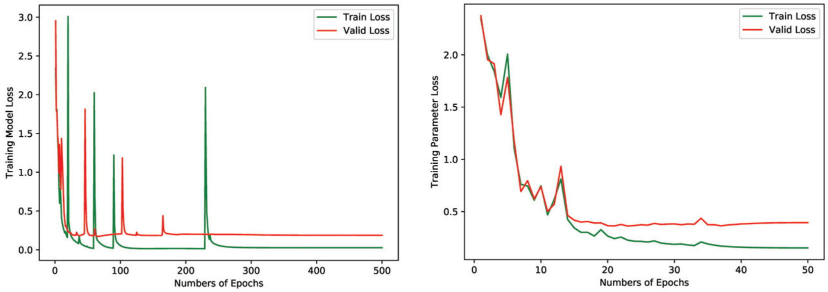
### *5.2. Settings*

**Baselines.** To compare our proposed mechanism with other existing methods, we consider the following baselines: Graph Convolutional Network (GCN) [3], Graph Attention Network (GAT) [18], Simplified Graph Convolution Network (SGC)) [4], JKNet [27], Multi-layer Perceptron (MLP) [45], Graph Sample and Aggregate (GraphSage) [46], DAGNN [47], GCNII [32], DenseGCN [48], and ResGCN [48].

**Configurations.** Our experiment is run on a NVIDIA GTX 3090Ti Graphical Card using PyTorch (version 1.7). In our experiment, GCN [3] is used as the baseline model, identity mapping is added to the convolution, and the data-driven propagation mechanism is used to obtain the network model. In all the experiments, we set the depth in {2, 4, 8, 16, 32, 64}. Throughout the experiment, we use the Adam optimizer [49]. We adopt the learning rate to be 0.005 and the maximum number of epochs to be 1000. We set the dropout to be 0.5, the dimensions of the hidden features to be 32, and the weight decay to be 0.001. We add L2 regularization to the model parameters. We set $\kappa_\ell = 1$ and $\beta_\ell = \log\left(\frac{\lambda}{\ell}+1\right) \approx \frac{\lambda}{\ell}$. The principle of setting $\kappa_\ell$ is to ensure the decay of the weight matrix adaptively increases as we stack more layers.

### *5.3. Results*

**Network evaluation.** We evaluate the training performance of the model by observing the training of the model in network structure selection and corresponding weight generation. The training results are shown in Figure 3a,b. We adaptively learn the network structure using the validation set and train and optimize the $\alpha$ in Equation (9) to obtain the network structure with the best performance. The jumping case of the training loss in Figure 3a is the process of optimizing the network architecture. After the adaptive learning of the network structure is completed, we use the method of training GNN to train and optimize the network parameters $w$ to obtain the final network model. Due to the use of differentiable methods for optimization, both our network structure selection training and parameter training can converge quickly. These training results confirm that the differentiable method we use is feasible and effective.



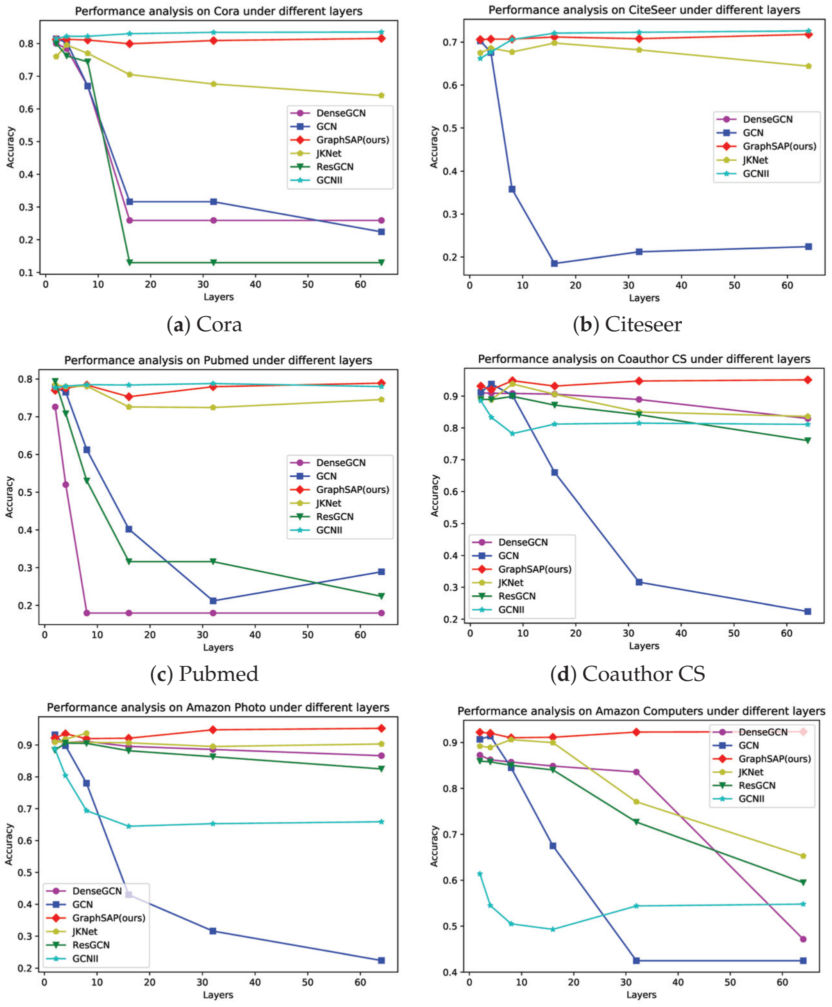
(**a**) (**b**)

**Figure 3.** The training state of our model on the Amazon Photo dataset [44] with a 64-layer network structure. (**a**) is the validation loss and training loss for the training selection network structure; (**b**) is the validation loss and training loss for training parameters corresponding to the network structure.

**Performance comparison.** The quantitative comparison results of node classification performance with other methods on various datasets are shown in Table 2. All results used for comparison are the best results achievable using the respective models. Our network achieves good performance on all seven datasets, achieving the highest classification accuracy on five of them. GAT shows good results on some datasets, such as Cora, but the effect on the Amazon Com dataset is mediocre. Compared with some current deep models GCNII, ResGCN, and DenseGCN, GCNII performs best on Cora and Citeseer datasets, but our network achieves the best results on all other datasets. In general, our model can be applied to various datasets and has achieved good results, which proves the effectiveness of the model. Our method also provides a feasible direction to better utilize the representational power of GNNs.

To investigate the model performance trends at different depths, we further compare the representational capabilities of our proposed model and existing models at different depths. The detailed comparison results of models with different depths are shown in Figure 4. From these experimental results, we can make the following observations. The baseline model (GCN) struggles to maintain consistent performance as we stack more layers. We also found that residual and dense connections can help improve the model performance on most datasets but not much for Amazon Computer and Pubmed datasets.

The Jumping Knowledge (JK) mechanism outperforms the baseline model (GCN) [3] in most cases. However, increasing depth also causes its performance to degrade. The GCNII model outperforms GCN and JKNet on multilayer networks, and the problem of over-smoothing is alleviated with increasing depth. However, GCNII performs poorly on four new datasets, and its generalizability is questionable. These experimental results further confirm that our proposed method is effective and feasible for training models with excellent representation ability.



(**a**) Cora (**b**) Citeseer

(**c**) Pubmed (**d**) Coauthor CS

(**e**) Amazon Photo (**f**) Amazon Computers

**Figure 4.** The performance comparison of the network we designed under different layers. We have performed experiments on different datasets. As can be seen in the figure, our data-driven layer connection learning method has relatively good network performance when the number of layers increases.

**Table 2.** Comparison of GraphSAP with other models for node classification tasks on Cora, Citeseer, PubMed, Coauthor CS, Coauthor Physics, Amazon Computers, and Amazon Photo datasets.

| Dataset<br>Model | Cora | Citeseer | PubMed | Coauthor CS | Coauthor Physics | Amazon Computers | Amazon Photo |
|---|---|---|---|---|---|---|---|
| GCN [3] | 81.2 | 70.3 | 77.8 | 93.8 | 96.1 | 90.7 | 92.2 |
| GAT [18] | 81.5 | 71.4 | 78.7 | 90.5 | 92.5 | 78.0 | 85.7 |
| SGC [4] | 79 | 68.5 | 76 | 83 | 90.5 | 88.5 | 90.3 |
| JKNet [27] | 79.6 | 69.8 | 78.4 | 93.8 | 97.1 | 90.5 | 93.6 |
| MLP [45] | 58.2 | 59.1 | 70.0 | 88.3 | 88.9 | 44.9 | 69.6 |
| GraphSage [46] | 76.6 | 67.5 | 76.1 | 85.0 | 90.3 | / | 90.4 |
| DAGNN [47] | 82.5 | 71.2 | 78.8 | 92.1 | 93.7 | 88.7 | 93.9 |
| GCNII [32] | **82.8** | **72.6** | 78.8 | 88.53 | 94.8 | 61.4 | 88.8 |
| Dense-GCN [48] | 80 | / | 72.6 | 93.6 | 96.1 | 91.1 | 94.1 |
| Res-GCN [48] | 80 | / | 78.4 | 93.8 | 96.0 | 91.7 | 94.3 |
| GraphSAP(ours) | 81.5 | 71.7 | **78.9** | **95.1** | **97.2** | **92.3** | **95.2** |

**Model Visualization.** We visualize the network structure learned by the model for node classification tasks on the Amazon Photo dataset, as shown in Figure 5. The network structure diagram shows that the final classification result is an aggregation of neighbors from different layers. Neighbors that need to be aggregated are adaptively learned by our method without relying on the manual design. The aggregation between layers of the network structure is irregular. Our method is flexible and widely applicable and has excellent graph representation ability.

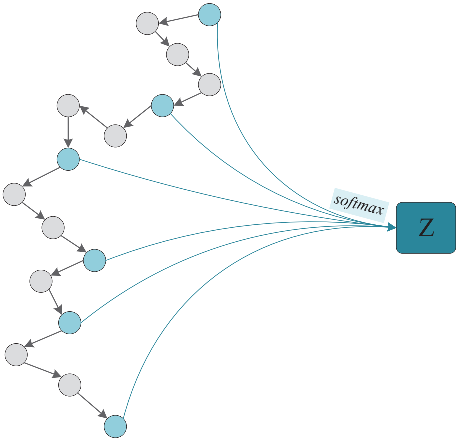


**Figure 5.** The 16-layer network structure learned by the model for the node classification task on the Amazon Photo dataset, where $Z$ denotes the final representation of the node after softmax.

## 6. Conclusions

We propose a data-driven propagation mechanism that adaptively learns different connections between different layers, i.e., learns combinations of different neighbors. This mechanism can alleviate the information redundancy and over-smoothing problems caused by the previously hand-designed GNN layer-connected architecture. Compared with other mainstream methods, the network architecture can be adapted to a variety of different datasets. The proposed GraphSAP achieves good performance on all three public datasets and achieves the best results on one of the public datasets as well as the new four datasets tested. In addition, our method has almost no performance degradation when the number of model layers is deepened. Further, the training efficiency is improved by adopting a more efficient differentiable learning algorithm.

In the future, we will explore more automatic learning methods to further improve the performance of GraphSAP. It also includes exploring other layer aggregators and studying the impact of different combinations of different layers and node aggregators on the graph structure. Furthermore, we can also explore tasks other than node classification tasks, such as graph classification.

**Author Contributions:** Conceptualization, Y.W.; methodology, Y.W.; validation, X.F.; writing—original draft preparation, Q.G. and X.H.; writing—review and editing, Y.W. and X.H.; supervision, W.M.; funding acquisition, Y.W. 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 (62276200, 62036006), the Natural Science Basic Research Plan in Shaanxi Province of China (2022JM-327) and the CAAI-Huawei MINDSPORE Academic Open Fund.

**Data Availability Statement:** The data used to support the findings of this study are available from the corresponding author upon request.

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

## References

1. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Netw.* **1995**, *3361*, 110–124.
2. Bronstein, M.M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Process. Mag.* **2017**, *34*, 18–42. [CrossRef]
3. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017; pp. 3031–3041.
4. Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; Weinberger, K. Simplifying Graph Convolutional Networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6861–6871.
5. Yao, L.; Mao, C.; Luo, Y. Graph Convolutional Networks for Text Classification. *AAAI Conf. Artif. Intell.* **2019**, *33*, 7370–7377. [CrossRef]
6. Hang, M.; Neville, J.; Ribeiro, B. A Collective Learning Framework to Boost GNN Expressiveness for Node Classification. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 4040–4050.
7. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1263–1272.
8. Zhang, J.; Dong, B.; Philip, S.Y. Fakedetector: Effective fake news detection with deep diffusive neural network. In Proceedings of the IEEE International Conference on Data Engineering, Virtual, 20–24 April 2020; pp. 1826–1829.
9. Jin, W.; Yang, K.; Barzilay, R.; Jaakkola, T. Learning multimodal graph-to-graph translation for molecular optimization. In Proceedings of the International Conference on Learning Representations, Vancouver, Canada, 30 April–3 May 2018; pp. 1536–1547.
10. Fout, A.; Byrd, J.; Shariat, B.; Ben-Hur, A. Protein interface prediction using graph convolutional networks. In Proceedings of the International Conference on Neural Information Processing Systems, Guangzhou, China, 14–18 October 2017; pp. 6533–6542.
11. Fischer, K.; Simon, M.; Olsner, F.; Milz, S.; Gross, H.M.; Mader, P. StickyPillars: Robust and Efficient Feature Matching on Point Clouds Using Graph Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2021; pp. 313–323.
12. Qi, X.; Liao, R.; Jia, J.; Fidler, S.; Urtasun, R. 3d graph neural networks for rgbd semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5199–5208.
13. 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**, *61*, 1–18. [CrossRef]
14. 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]
15. Wu, Y.; Ding, H.; Gong, M.; Qin, A.K.; Ma, W.; Miao, Q.; Tan, K.C. Evolutionary Multiform Optimization with Two-stage Bidirectional Knowledge Transfer Strategy for Point Cloud Registration. *IEEE Trans. Evol. Comput.* **2022** . [CrossRef]
16. Kosaraju, V.; Sadeghian, A.; Martín-Martín, R.; Reid, I.; Rezatofighi, S.H.; Savarese, S. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1–14.
17. Zheng, C.; Fan, X.; Wang, C.; Qi, J. Gman: A graph multi-attention network for traffic prediction. *Aaai Conf. Artif. Intell.* **2020**, *34*, 1234–1241. [CrossRef]
18. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, Canada, 30 April–3 May 2018; pp. 1420–1431.

19. Kersting, K.; Kriege, N.M.; Morris, C.; Mutzel, P.; Neumann, M. Benchmark Data Sets for Graph Kernels. 2016. Available online: http://www.graphlearning.io/ (accessed on 6 February 2020).
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
21. Namata, G.; London, B.; Getoor, L.; Huang, B.; EDU, U. Query-driven active surveying for collective classification. *Int. Workshop Min. Learn. Graphs* **2012**, *8*, 1–8.
22. Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; Eliassi-Rad, T. Collective classification in network data. *Artif. Intell. Mag.* **2008**, *29*, 93. [CrossRef]
23. Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. In Proceedings of the IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 10–15 July 2005; pp. 729–734.
24. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [CrossRef] [PubMed]
25. Xu, D.; Ruan, C.; Korpeoglu, E.; Kumar, S.; Achan, K. Inductive representation learning on temporal graphs. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020; pp. 890–902.
26. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
27. Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.i.; Jegelka, S. Representation learning on graphs with jumping knowledge networks. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5453–5462.
28. He, X.; Zhao, K.; Chu, X. AutoML: A Survey of the State-of-the-Art. *Knowl.-Based Syst.* **2021**, *212*, 106622–106649. [CrossRef]
29. Liang, J.; Meyerson, E.; Hodjat, B.; Fink, D.; Mutch, K.; Miikkulainen, R. Evolutionary Neural AutoML for Deep Learning. In Proceedings of the Genetic and Evolutionary Computation Conference, Prague, Czech Republic, 13–17 July 2019; pp. 401–409.
30. 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]
31. Zoph, B.; Le, Q.V. Neural architecture search with reinforcement learning. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017; pp. 3075–3088.
32. Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; Li, Y. Simple and deep graph convolutional networks. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 1725–1735.
33. Spinelli, I.; Scardapane, S.; Uncini, A. Adaptive Propagation Graph Convolutional Network. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4755–4760. [CrossRef] [PubMed]
34. Sun, Y.; Yao, X.; Bi, X.; Huang, X.; Zhao, X.; Qiao, B. Time-Series Graph Network for Sea Surface Temperature Prediction. *Big Data Res.* **2021**, *25*, 100237. [CrossRef]
35. Bi, X.; Liu, Z.; He, Y.; Zhao, X.; Sun, Y.; Liu, H. GNEA: A Graph Neural Network with ELM Aggregator for Brain Network Classification. *Hindawi Publ. Corp.* **2020**, *2020*, 8813738. [CrossRef]
36. Fan, X.; Gong, M.; Tang, Z.; Wu, Y. Deep Neural Message Passing with Hierarchical Layer Aggregation and Neighbor Normalization. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *25*, 540–554. [CrossRef] [PubMed]
37. Fan, X.; Gong, M.; Wu, Y.; Qin, A.K.; Xie, Y. Propagation Enhanced Neural Message Passing for Graph Representation Learning. *IEEE Trans. Knowl. Data Eng.* **2021**. [CrossRef]
38. Klicpera, J.; Bojchevski, A.; Günnemann, S. Predict then propagate: Graph neural networks meet personalized pagerank. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019; pp. 1314–1325.
39. Xiao, T.; Chen, Z.; Wang, D.; Wang, S. Learning How to Propagate Messages in Graph Neural Networks. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Virtual, 14–18 August 2021; pp. 2235–2246.
40. Colson, B.; Marcotte, P.; Savard, G. An overview of bilevel optimization. *Ann. Oper. Res.* **2007**, *153*, 235–256. [CrossRef]
41. Maclaurin, D.; Duvenaud, D.; Adams, R. Gradient-based hyperparameter optimization through reversible learning. In Proceedings of the International Conference on Machine Learning, Lile, France, 6–11 July 2015; pp. 2113–2122.
42. Pedregosa, F. Hyperparameter optimization with approximate gradient. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 737–746.
43. Franceschi, L.; Frasconi, P.; Salzo, S.; Grazzi, R.; Pontil, M. Bilevel programming for hyperparameter optimization and meta-learning. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1568–1577.
44. McAuley, J.; Targett, C.; Shi, Q.; Van Den Hengel, A. Image-based recommendations on styles and substitutes. In Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 43–52.
45. Leshno, M.; Lin, V.Y.; Pinkus, A.; Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw.* **1993**, *6*, 861–867. [CrossRef]
46. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive representation learning on large graphs. *Int. Conf. Neural Inf. Process. Syst.* **2017**, *152*, 1025–1035.
47. Liu, M.; Gao, H.; Ji, S. Towards Deeper Graph Neural Networks. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 23–27 August 2020; pp. 5221–5233

48. Li, G.; Muller, M.; Thabet, A.; Ghanem, B. DeepGCNs: Can GCNs Go as Deep as CNNs? In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 9267–9276.
49. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015; pp. 3612–3624.

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

*electronics*

MDPI

*Article*

# CM-NET: Cross-Modal Learning Network for CSI-Based Indoor People Counting in Internet of Things

**Jing Guo [†], Xiaokang Gu [†], Zhengqi Liu, Minghao Ji, Jingwen Wang, Xiaoyan Yin and Pengfei Xu [*]**

School of Information Science and Technology, Northwest University, Xi'an 710000, China

[*] Correspondence: fpxu@nwu.edu.cn

[†] These authors contributed equally to this work.

**Abstract:** In recent years, multiple IoT solutions have used computational intelligence technologies to identify people and count them. WIFI Channel State Information (CSI) has recently been applied to counting people with multiple benefits, such as being cost-effective, easily accessible, free of privacy concerns, etc. However, most current CSI-based work is limited to human location-fixed environments since human location-random environments are more complicated. Aiming to fix the problem of counting people in human location-random environments, we propose a solution using deep learning CM-NET, an end-to-end cross-modal learning network. Since it is difficult to count people with CSI straightforwardly, CM-NET approaches this problem using deep learning, utilizing a multi-layer transformer model to automatically extract the correlations between channels and the number of people. Owing to the complexity of human location-random environments, the transformer model cannot extract characteristics describing the number of people. To enhance the feature learning capability of the transformer model, CM-NET takes the feature knowledge learned by the image-based people counting model to supervise the learning process. In particular, CM-NET works with CSI alone during the testing phase without any image information, and ultimately achieves sound results with an average accuracy of 86%. Meanwhile, the superiority of CM-NET has been verified by comparison with the latest available related methods.

**Keywords:** people counting; CSI; knowledge distillation; cross-modal learning network; computational intelligence

**Citation:** Guo, J.; Gu, X.; Liu, Z.; Ji, M.; Wang, J.; Yin, X.; Xu, P. CM-NET: Cross-Modal Learning Network for CSI-Based Indoor People Counting in Internet of Things. *Electronics* **2022**, *11*, 4113. https://doi.org/10.3390/electronics11244113

Academic Editor: Yue Wu

Received: 13 November 2022
Accepted: 6 December 2022
Published: 9 December 2022

## 1. Introduction

People counting provides key information for a wide range of services and applications, such as crowd control for places of interest and marketing research for malls. However, human behavior can be unpredictable; thus, people counting may encounter various challenges, such as object occlusions, pedestrian overlaps, and demands for real-time processing. Traditional solutions to these issues fall roughly into the following categories: environmental sensor-based, vision-based, and wireless signal-based methods.

Along with the advances in sensing technology, many sensor-based networks provide rather good accuracy in estimating the number of people by analyzing variations in the surroundings, such as temperature [1], sound [2], and carbon dioxide [3]. However, the feasibility of sensor-based counting methods is hindered by constraints such as expensive equipment, the complexity of the operation, and limited scope. Vision-based methods have been widely used in many public places [4,5], yet these methods are inherently flawed. First, cameras work only in a line-of-sight pattern, leaving many areas blind to monitoring. Second, smoke or a lack of light in the environment will severely degrade the image quality. Thirdly, overlapping objects further hamper the model's performance. Wireless signals based methods perform people counting based on radio-frequency signals (WIFI [6–9], UWB radar [10,11], etc.). The advantage of these methods is that they are not affected by light, do not violate human privacy and can achieve good recognition results.

For indoor scenarios, the WIFI-based methods have natural advantages: (1) No additional devices are required. With the widespread deployment and coverage of WIFI in public places or home, the existing infrastructures provide the data base for the WiFi-based methods. (2) The monitored person is not required to carry any smart devices or sensors. The principle of the WiFi-based methods is the WIFI signal will reflect, diffract and refract when encountering obstacles or moving individuals in the process of propagation, to form a regular signal change pattern. In addition, the number of people can be identified by analyzing the signal change pattern.

The most commonly used WIFI-based methods are the coarse-grained Received Signal Strength (RSS) and the fine-grained Channel State Information (CSI). For the former, while RSS data can be easily obtained for most off-the-shelf wireless devices, RSS measurements are affected by multipath fading and environmental noises. The RSS dataset is often expanded to improve the count accuracy, which inevitably incurs additional labor and time costs. A recently proposed solution [9] to the above problem estimates the number of people using CSI, which can provide richer information than RSS, including each subcarriers amplitude and phase information. Meanwhile, CSI is more sensitive to the number of people in the environment than RSS. Present-day works, however, assume all human objects are fixed, which is not the case in practice. Therefore, it cannot be applied to scenarios involving random changes in human positions.

A technical challenge that needs to be solved to turn the idea into a working system, i.e., how to extract good features characterizing the number of people in the conditions of different locations. Inspired by knowledge distillation [12], we guide the CSI-based network by other modal networks acting as teachers so that the CSI-based network can extract useful features from the crowd. Several existing methods of counting people have been compared and analyzed, including those using visible images, infrared images, and other environmental sensing information. It has been found that image-based methods are easy to implement and can achieve high accuracy under normal conditions. Therefore, we propose a cross-modal learning network CM-NET, which uses the image-based network as a teacher network to guide the learning of the student network (CSI-based network). As a result of CM-NET, CSI-based networks are capable of achieving more accurate results and alleviating degradation in accuracy caused by the change in human locations.

In CM-NET, both the teacher network and the student network require training. The teacher network is first pre-trained on the COCO dataset and then fine-tuned on our data. The student network is trained using the feature knowledge learned from the trained teacher network. CM-NET avoids privacy issues by using only image information in training and only CSI data in testing.

The main contributions of this paper are as follows:

1. For the first time, we use multimodal knowledge distillation to perform CSI-based people counting. We use multimodal knowledge to compensate for the limits of unimodal network that need a lot of training data and weak feature representation, providing a new idea for CSI-based people counting methods.

2. We propose a cross-modal learning network, CM-NET, which uses the feature information output by the vision network as a supervisory signal to guide feature learning in CSI-based network, alleviating the performance degradation of CSI-based network due to location changes.

3. To the best of our knowledge, we build the first international dataset of indoor multimodal (video and CSI), which includes multimodal data from 60 different locations and 9 different numbers of people. On this dataset, the CM-NET proposed in this paper achieves an average accuracy of 86%, which is better than other current methods for counting people.

The rest of the paper is structured as follows. Section 2 provides an overview and analysis of existing methods and networks related to people counting. Section 3 describes our proposed people counting method more detailed. Section 4 gives the relevant experimental results and provides a detailed description and analysis of the results. Finally, Section 5 concludes the paper.

## 2. Related Work

Current methods can be classified into three categories: Environmental sensor-based methods, Vision-based methods and Wireless signal-based methods. In this section, we will discuss the existing works on the above three types of people counting methods.

**(1) Environmental sensor-based methods.** These methods can be broadly classified into the following three categories based on the type of sensor: passive infrared sensor-based, sound sensor-based, and carbon dioxide sensor-based.

Passive infrared sensors infer the number of moving bodies based on the ambient temperature change caused by moving objects in the sensor area. F. Wahl et al. [1] arranged passive infrared sensors at each pedestrian walkway and then designed a probabilistic distance-based algorithm to estimate the number of people in the environment. The test results show that the probabilistic distance-based algorithm can compensate for the error due to the infrared mask effect. The statistical error increases significantly when multiple people simultaneously pass through the pedestrian passage. Statistical errors can also be caused by interference from sunlight and heat.

The sound sensor-based people counting method sends an ultrasonic signal to the monitoring area using a sound sensor. It reflects when the ultrasonic signal encounters a moving object, such as a person's body. When the sound sensor receives the reverberation of the transmitted signal, the number of people in the environment can be inferred based on characteristics such as reception time or signal attenuation. O. Shih et al. [2] designed a system to count people in an area using changes in the room's acoustic properties. Experiments have shown that the acoustic sensor-based solution performs better in smaller indoor spaces and crowds. However, the system's scalability is somewhat limited, as the accuracy decreases significantly with increasing space and occupancy. In addition, the number of people estimates may be distorted due to the presence of a large amount of sound-absorbing materials often present in indoor environments.

Since people's respiration produces carbon dioxide in a room, it is possible to infer the number of people by the concentration of carbon dioxide. Such scheme uses sensors to measure the concentration of carbon dioxide in a room and estimate the number of people. H. Rahman et al. [3] proposed a people counting system based on indoor carbon dioxide emissions. Experiments show that the carbon dioxide sensor-based people counting method is slow to respond to dynamic changes as it takes a certain amount of time for the carbon dioxide concentration to change if someone enters or leaves the room.

**(2) Vision-based people counting methods.** N. Dalal et al. [4] proposed a technique for intact human body-based detection. This technique extracts directional gradient histograms from pedestrian sample images as features for person counting using a linear support vector machine [5] for classification. However, intact body-based detection techniques are only suitable for cases where bodies do not overlap, so the system is vulnerable to occlusions or complex backgrounds, resulting in poor recognition performance. Wu Bo et al. [13] proposed a partial body-based detection technique, which uses part of the pedestrian's body structure to perform detection. For example, the Adaboost [14] network can be trained on the head and shoulders of pedestrians. Then, the number of pedestrians' heads and shoulders are detected in the image to calculate the number of people. Although this technique can reduce statistical errors caused by multi-person overlap, it still poses problems determining image region division and sliding window size.

The regression network-based people counting technique [15] using a proven regression network counts the number of people in an image by considering the crowd in the image as a whole object. The technique extracts foreground image features from the collected images [15], then trains a regression model using the obtained feature set, and then uses the built regression model to obtain the headcount information from the test sample. The features used can be classified as foreground image features, edge features [16], gradient features [17], and texture features [18]; regression strategies comprise Gaussian process regression [19], nonlinear regression, and neural networks; images can be processed according to partitioning, sliding windows, global estimation, etc. Unlike detection-based algorithms, regression networks have good generalization capabilities, excellent portability, and endless possibilities. A. Davies et al. [15] found a correlation between the number of people in the environment and the foreground pixel area and chose to use regression information to identify the number of pedestrians in an image. This work was the first to introduce regression methods to crowd analysis and indirectly inspired people counting. Regression-based methods also fail to overcome the impact of overlap occlusion of the human body.

**(3) Wireless signal-based methods.** Wireless signal-based methods use deployed wireless devices to build an environment-aware system for people counting. These techniques not only compensate for the vulnerability of sensor-based people counting techniques to environmental influences but also fill the gap of video-based counting techniques that do not work in privacy and obscuration situations. As far as the current research in this field is concerned, it can be divided into the following three classes based on the characteristics of the chosen radio signal: UWB signal-based people counting methods, RSS-based people counting methods, and CSI-based people counting methods.

People counting based on UWB signals [10] is mainly performed through radar networks. The method first removes static objects and people reflections by the background phase cancellation method [11]. After that, it calculates the number of people based on the waveform characteristics of the received signal. J. W. Choi et al. [20] proposed an IR-UWB radar system that sends a broadband pulsed signal along with receiving the backscattered signal from the environment to infer the number of human targets. Experiments have shown that the system can detect up to three targets, with errors reaching 8%. UWB signal-based counting technology has excellent recognition performance; however, this technology requires expensive and special hardware devices to support it, so it is difficult to prevail in the production life of people.

In 2008, M. Nakatsuka et al. [6] first validated the feasibility of using RSSI for people counting and designed a linear regression-based model. This model relies mainly on the mean change of RSSI measurements to calculate the number of people. Experimental results demonstrated that the RSS between two radio nodes decreases as the number of people bodies located between these nodes increases, and they proposed linear network can detect up to 9 people. In 2015, T. Yoshida et al. [7] implemented an RSS-based people counting technique based on support vector regression. This method can count up to 7 people with an accuracy of up to 77%. Alsamhi et al. [21] proposed extending the ANN's method to UAVs to predict the intensity of signals in different areas of the city, planning for WIFI-based outdoor people counting in the future. Although RSSI-based people counting technology can significantly reduce equipment deployment costs by depending on WiFi infrastructure, RSSI, as a coarse-grained description of the channel, can only reflect the signal's fading after propagation. In a complex and noisy indoor environment, the performance of a system designed by RSS can be greatly compromised.

With the development of technology and the unremitting efforts of researchers, D. Halperin et al. [8] extracted Channel State Information (CSI) on commercial Wi-Fi supporting 802.11n protocol using the self-developed CSI Toolkit. With the successful extraction of CSI, Xi Wei et al. [9] proposed a passive number counting system based on CSI. In this system, the authors utilize the percentage of non-zero elements in the inflated CSI matrix as features to describe the variation of the wireless channel, and use the gray Verhulst model for feature training thereby building a library of feature-number relations, which can be used to identify the number of people at the testing phase. S. D. Domenico et al. [22] proposed a people counting system based on differential CSI measurements. The system uses Euclidean distance to represent the difference between two CSI measurements and uses features such as first- and second-order statistical moments for classification. Experiments show that the system can accommodate up to seven people in an indoor environment with a classification error of about 15%. Compared with the RSSI-based counting method, the CSI-based method offers more stable performance and relatively higher accuracy. In 2016, Alsamhi et al. [23] proposed a technique for maintaining signal quality on high-altitude mobile platforms, improving the signal quality and mitigating the impact of signal quality on the number of people counted.

## 3. Design and Training of CM-NET

Generally, the performance of visual-based people counting models, once trained, is almost position-independent. However, visual-based people counting models may be susceptible to factors such as illumination and occlusion, leading to significant performance degradation. Furthermore, privacy concerns involving those being watched may arise with vision-based networks. In contrast, CSI-based networks rely on CSI to infer the number of people, which is readily available even in low-light settings. These methods have very few privacy concerns because they do not specifically use the monitored people's images. Nevertheless, monitored individuals' location changes decrease CSI-based networks' performance. The location sensitivity limits of CSI-based people counting jobs may be partially overcome if CSI-based counting networks can achieve comparable anti-interference and discriminating skills to vision-based networks [24,25].

Based on the preceding study and motivated by the knowledge distillation technique, we propose an end-to-end cross-modal learning network called CM-NET that trains the CSI-based network using the class probabilities from the vision-based network as soft labels. CM-NET's architecture consists of teacher and student networks; the training framework is depicted in Figure 1. CM-NET's teacher network performs people counting with visible images.

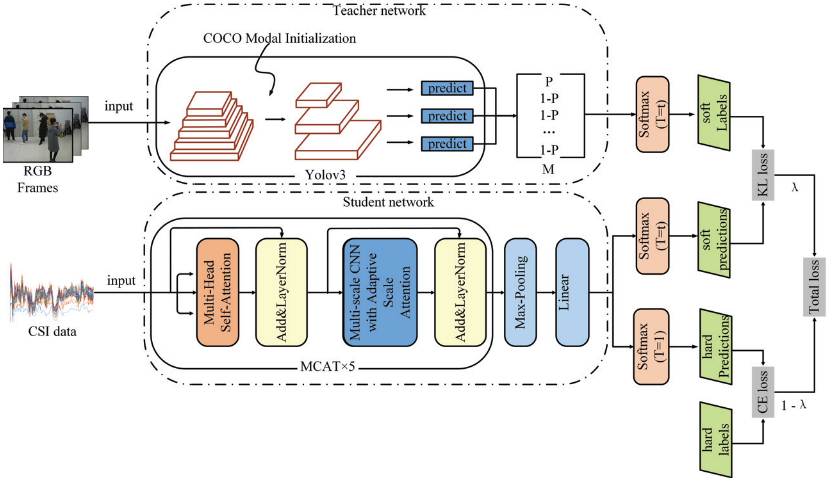


**Figure 1.** The training framework of CM-NET. It has two components: the teacher's training network and the student's training network. The teacher is a vision-based people counting network, whose network is first initialized using the trained weights from the COCO dataset and then fine-tuned with our collected video data. The student is a CSI-based people counting network, whose network training process is supervised by the soft labels from the teacher network.

In CM-NET's teacher network, the Yolov3 target detection module first obtains the probability of each person in the image. Then, the probability matrix of the total number of people category is generated by the uniform distribution. Finally, the probability matrix is fed to the softmax classifier. The output of the softmax classifier is used as a soft label to guide the training of the student network. The student network of CM-NET uses CSI to count people, consisting of the transformer feature extraction module and the softmax classifier module. It should be noted that CM-NET only introduces the soft labels generated by the teacher network as supervised information during the student network's training process.

Section 3.1 describes the teacher network of CM-NET, and Section 3.2 describes the student network of CM-NET. Finally in Section 3.3, we will describe the design of the CM-NET loss function.

### *3.1. Teacher Network: Vision-Based People Counting Network*

CM-NET's teacher network uses Yolov3 as the detection model. We customized the output module of Yolov3 in order to transfer the knowledge gained to the student network. We statistically analyze the degree of confidence of each individual in the Yolov3 photos, denoted as $p_i$, where $i$ represents the $i$th people detected. In addition, the statistics are summed and averaged to obtain the confidence scores for the total number of people denoted as $P$, the corresponding function is shown in Equation (1). Then, the confidence score matrix for total number of people class denoted as $M = [m_1, m_2, m_3, \ldots, m_K]$, was obtained by expanding $P$ through a label smoothing technique, the corresponding function is shown in Equation (2). Finally, $M$ is fed into the softmax layer to obtain the prediction results.

$$P = \sum_{i}^{n} p_i / n \tag{1}$$

$$m_j = \begin{cases} P & \text{if } j = n \\ \frac{1-P}{K-1} & \text{if } j \neq n \end{cases} \quad (j = 1, 2, 3, \ldots, K)\cdot \tag{2}$$

where $K$ represents the total classes of crowd. It is important to note that the soft labels generated by the teacher network can more accurately represent the similarity across classes than the actual labels. The knowledge distillation approach adds a temperature coefficient to the original softmax function to soften the class probabilities in order to better represent this similarity between classes. The softened soft label is represented as $\mathrm{O}_V = [v_1, v_2, v_3, \ldots, v_K]$, and it is calculated as shown in Equation (3).

$$v_i = \frac{e^{(m_i/T)}}{\sum_{k=1}^{K} e^{(m_k/T)}} \tag{3}$$

where it can be seen that the distribution of probability output is "softer" under high temperatures. The "softer" probability distribution is more effective in extracting knowledge and training the network.

### *3.2. Student Network: CSI-Based People Counting Network*

CSI is 2D time series data denoted as $C \subseteq \mathbb{R}^{d \times n}$, where $d$ and $n$ represents channel number and received package number, separately. The CSI data contains noise and outliers due to the presence of multipath effects and ambient noise in the room, as well as defects in the hardware equipment itself. If the raw data are used directly, the performance of CSI-based people counting networks will be significantly degraded. For this reason, we performed the necessary preprocessing. We first apply Equation (4) to remove the outliers from the raw CSI data and then smooth CSI data using linear interpolation.

$$C = [\mu - \eta \times \delta, \mu + \eta \times \delta] \tag{4}$$

where $\mu - \eta$ is the median of a set of observations, $\delta$ is the absolute deviation from the median, and $\eta$ is the empirical constant, which is taken as 2 in this paper.

Then, we use a soft-threshold denoising method based on wavelet analysis to eliminate the noise. In this paper, we use a Gaussian function as the wavelet basis function for 12-layer denoising of CSI data and a soft thresholding method for high frequency coefficients. The processed CSI data are fed into the student network for training. The student network uses the transformer framework as the base network, which consists of five MCAT layers [26], a pooling layer and a fully connected layer. The student network feeds the softmax layer with the people in the crowd extracted by the transformer's digit extraction algorithm to obtain the prediction results. It is worth noting that during the training process, the predictions of the student network are divided into hard predictions denoted as $O_S = [s_1, s_2, s_3, \ldots, s_K]$ and soft predictions denoted as $O'_S = \left[s'_1, s'_2, s'_3, \ldots, s'_K\right]$. The $O_S$ generated by the student network uses the original softmax. This is expressed in Equation (5):

$$s_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \tag{5}$$

where $z_i$ is the output of the fully connected layer of the student network.

Similar to $O_V$ generated by the teacher network, $O'_S$ generated by the student network uses the temperature coefficients of the softmax layer, as shown in Equation (6).

$$s'_i = \frac{e^{(z_i/T)}}{\sum_{k=1}^{K} e^{(z_k/T)}} \tag{6}$$

### *3.3. Design of the Loss Function*

The loss function $L$ of CM-NET consists of two components: the KL scatter loss $L_{kl}$ between the output of the student network and the output of the teacher network, and the cross-entropy loss $L_{ce}$ between the output of the student network and the hard label, where the hard label is the true label of the sample noted as $Y$. The KL scatter loss $L_{kl}$ can be formulated as follows:

$$L_{kl} = \sum O_{S'} \log \frac{O_{S'}}{O_V} \tag{7}$$

the cross-entropy loss $L_{ce}$ can be formulated as follows:

$$L_{ce} = \sum O_V \log Y + (1 - O_S) \log(1 - Y) \tag{8}$$

Finally, the loss function $L$ of CM-NET can be formulated as follows:

$$\min_S L = \lambda L_{kl} + (1 - \lambda) L_{ce} \tag{9}$$

where $\lambda$ is the balance factor. In the training process of CM-NET, we fixed the parameters of the teacher network and updated the student network using the gradient descent method. The student and teacher networks are dropout-regularized during the training process to avoid overfitting.

## 4. Experiments

The experimental section begins with Section 4.1, which describes how the equipment is deployed and data collected. In Section 4.2, we describe the preprocessing details of the dataset and the CM-NET hyperparameter settings. Section 4.3 describes the metrics for evaluating CM-NET's performance and counting the number of people. Section 4.3.3 shows comparisons between our experiments and existing work.

### *4.1. Equipment Deployment and Data Collection Process*

Our data collection is based on commercially available hardware, including a TP-Link camera, wireless router, and laptop with an Intel 5300 NIC. The TP-link wireless router serves as the transmitter, and the laptop the receiver. The receiver and transmitter were mounted on a 0.75 m stand separated by 5.6 m. In order to make sure that the camera would cover the entire area used for collecting CSI data, we fixed the camera to the wall four meters high.

We set up the equipment mentioned above in our lab to gather CSI and video data. The size of the area where we collected CSI data was 2.4 m × 3.2 m, and the area is shown in Figure 2. We enlisted 9 volunteers for this experiment, including 2 females and 7 males.

We gathered information under two cases: fixed location (dataset-1) and non-fixed location (dataset-2). In detail, we asked different numbers of volunteers to stand once at each of the fixed 60 locations and collected CSI data lasting 15 s, and then added corresponding labels to the data for the category of each number of volunteers. Thus, we obtained 540 samples (9 Classes × 60 locations). For the second batch of data, we loosened the restriction by permitting volunteers to stand wherever within the CSI data collection area. In a similar number, we collected 540 samples and assigned each sample to the corresponding class of people. Specifically, the camera remained active during the whole data gathering process.

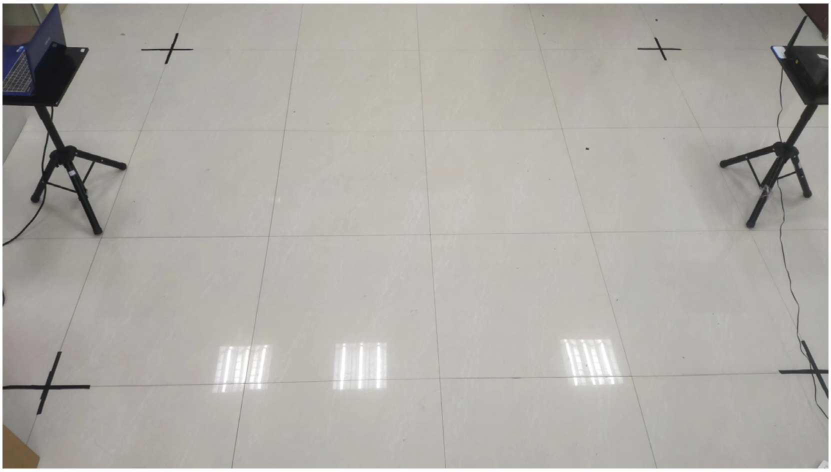


**Figure 2.** The area where the CSI data were collected. The crossbelt marks the boundary of the area, which is 2.4 m × 3.2 m in size.

### *4.2. Preprocessing Details of the Dataset and CM-NET Hyperparameter Settings*

We extract the CSI amplitude data from the CSI dataset and preprocess it with noise and outlier removal. After that, we segmented the CSI dataset based on a 3 s time window to expand the dataset. Since the location and number of people do not change in each video sample we collect, so we sample fixed interval frames from the original video data as our video dataset. Specifically, we conduct sampling every 3 s so that each CSI sample has a frame to correspond to. This provides the benefit of speeding up CM-NET training while conserving computational resources.

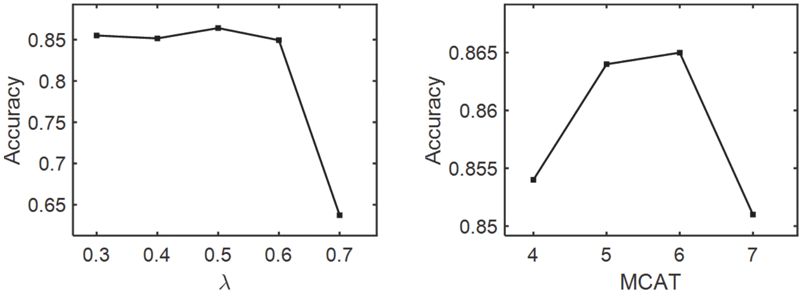
The teacher and student networks are trained as part of the CM-NET training procedure. We first use the COCO dataset to establish the initial training weights for the teacher network, and then we use the video dataset we collected to fine-tune them. The weight parameters of the teacher network are frozen once the training process is complete. Before feeding the CSI data into CM-NET synchronously with the corresponding video frames, we load the teacher network's weight parameters into it to train the student network. In the course of training the student network, the soft labels generated by the teacher network and the tagged real labels are used in conjunction to supervise the training process.

CM-NET training is performed on a laptop with an RTX 2060 GPU, based on the PyTorch deep learning framework. The batch size for training the student network is set to 8, the training parameters are optimized using the Adam optimizer, the learning rate is set to 0.001, the MCAT is set to 5 layers, and the balance factor $\lambda$ is set to 0.5. If not stated otherwise, the following results are assessed using a 7:1:2 random allocation for the training, testing, and validating datasets.

Performance of CM-NET under Different Parameters

(1) The impact of different $\lambda$. The magnitude of $\lambda$ determines how far the teacher network can affect the student network during training. In this paper, we selected four different degrees and tested them on dataset-1, and the results are shown in Figure 3a. The performance of CM-NET with five degrees is lighter (0.3): 0.8550, light (0.4): 0.8515, moderate (0.5): 0.8640, heavy (0.6): 0.8493, and heavier (0.7): 0.6374. 0.6374. Not surprisingly, the performance of the student network did not rise as the effect of the teacher network increased. Only appropriate instruction improves the ability of the student network to extract the characteristics of the number of people. Suppose the teacher network interferes too much with the learning of the student network. In that case, it will make the students depend too much on the teacher network's inference when extracting features, which leads to the student network not paying enough attention to its data, and the model's performance decreases.

(2) The impact of MCAT with different layers. The number of layers of MCAT determines the depth of the CM-NET network. In this paper, we selected four different depths and performed tests on dataset-1, whose results are shown in Figure 3b. The performance of CM-NET with four depths is four layers: 0.8540, five layers: 0.8640, six layers: 0.8654, and seven layers: 0.8512. Obviously, the depth of the network does affect the ability of CM-NET to extract features. For the same number of samples, an appropriate increase in the number of MCAT layers can boost the extraction ability of the network. However, increasing the number of MCAT layers without expanding the data can cause the network to overfit during the training process, which leads to worse prediction results. Notably, the final number of MCAT layers selected in this paper is five. This is because the accuracy of the MCAT of five layers is similar to that of six. However, five layers MCAT causes less training time, fewer parameters, and fits with less difficulty.



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

**Figure 3.** Performance of CM-NET under different parameters. (**a**) The impact of different $\lambda$. (**b**) The impact of MCAT with different layers.

## *4.3. Experimental Results*

### 4.3.1. Performance Evaluation Metrics

In this paper, four evaluation metrics of Accuracy, Recall, Precision, and F1score are used to evaluate the people counting performance of CM-NET, and the expressions of each evaluation metric are as follows:

$$Accuracy = \frac{TP}{FP + FN + TP + TN} \tag{10}$$

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

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$F1\text{-}score = 2\frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

where $TP$ represents the number of samples that were correctly classified by the classifier and rated positive by the classifier; $TN$ represents the number of samples that were correctly classified by the classifier and rated negative by the classifier; $FP$ represents the number of samples that were incorrectly classified by the classifier and rated positive by the classifier; and $FN$ represents the number of samples that were incorrectly classified by the classifier and rated negative by the classifier.

### 4.3.2. CM-NET Performance Evaluation

To test the performance of CM-NET at multiple locations, we use CM-NET and CM-NET (Training without image) on dataset-1 (shown in Table 1) and dataset-2(shown in Table 2), respectively.

**Table 1.** CM-NET performance on dataset-1.

| Model | Accuracy | Recall | Precision | F1-Score |
|---|---|---|---|---|
| CM-NET | 86.40% | 85.44% | 85.68% | 85.15% |
| CM-NET (Training without image) | 82.44% | 81.19% | 81.49% | 80.72% |

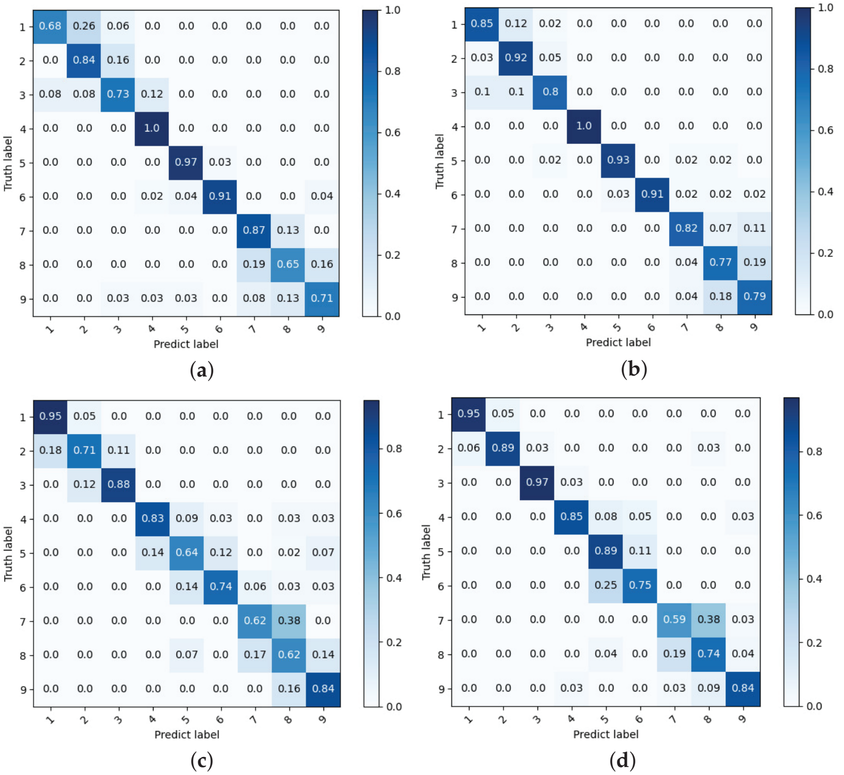
Table 1 compares the performance of CM-NET and CM-NET (Training without image) on dataset-1. On dataset-1, it is simple to observe that CM-NET (Training without image)'s people counting accuracy can only reach 82.44% and other metrics are around 81%. In contrast, CM-NET's accuracy can reach 86.40%, and other metrics are around 85%, which is superior to CM-NET (Training without image). By examining the test data, it can be seen that even if the position is fixed, the CSI amplitude is still subject to position interference. The CM-NET (Training without image) performs poorly in classification because it cannot extract more effective crowd features. In contrast, CM-NET was trained by introducing soft labels produced by the vision-based network. This allowed the network to extract a greater variety of people features, and its performance in counting was greatly enhanced.

**Table 2.** CM-NET performance on dataset-2.

| Model | Accuracy | Recall | Precision | F1-Score |
|---|---|---|---|---|
| CM-NET | 83.61% | 85.40% | 82.92% | 82.65% |
| CM-NET (Training without image) | 75.92% | 75.64% | 75.80% | 75.43% |

Table 2 compares the performance of CM-NET and CM-NET (Training without image) on dataset-2. We can see that on dataset-2, CM-NET (Training without image)'s accuracy in counting people can only reach 75.92%, and all other metrics are around 75%, while CM-NET's accuracy in counting people can reach 83.61%, and all other metrics are around 82%, which is superior to CM-NET (Training without image). The results shown in Table 2 indicate that location change interferes more with CSI when the location of the detected people is not fixed. As the location of detected people changes randomly, and we cannot collect data from all locations, the CM-NET (Training without image) network extracts information with significantly less attention being paid to the feature of the number of people, and its performance for counting people declines. While, the location change almost did not affect the performance of the vision-based network. CM-NET transfers the knowledge learned by the vision-based people counting network to the CSI-based network using the technique of knowledge distillation. By doing so, CSI-based networks can extract features more efficiently and count people more effectively.

We present the confusion matrix of CM-NET and CM-NET (Training without image) test results in Figure 3. Where Figure 4a shows the confusion matrix of CM-NET (Training without image) on dataset-1, Figure 4b the confusion matrix of CM-NET on dataset-1, Figure 4c the confusion matrix of CM-NET (Training without image) on dataset-1, and Figure 4d the confusion matrix of CM-NET on dataset-2.



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

**Figure 4.** Confusion matrix of CM-NET (Training without image) and CM-NET test results. In the confusion matrix, the row coordinates represent the number of people predicted by the network and the vertical coordinates represent the actual number of people; the numbers in the matrix represent the proportion of correctly predicted samples to the total samples in the class. (**a**) CM-NET (Training without image) confusion matrix on dataset-1. (**b**) CM-NET confusion matrix on dataset-1. (**c**) CM-NET (Training without image) confusion matrix on dataset-2. (**d**) CM-NET confusion matrix on dataset-2.

It is clear from the confusion matrix in Figure 4 that CM-NET increases the CSI-based network's accuracy for classifying several people-counting groups. The improved accuracy is attributed to the fact that CM-NET makes the CSI-based network pay more attention to the features of the number of people when extracting features by introducing soft labels of visual information as supervisory signals, which attenuates the effect of changes in location on the network feature extraction.

### 4.3.3. Comparative Experiments and Correlation Analysis with Existing Related Work

We compared with some existing networks, including DNN [27], SVM, KNN, and the current state-of-the-art classification network Two-Stream Transformer [24], to show the superiority of CM-NET in people counting at multiple places.

The performance of the five current networks on dataset-1 is shown in Table 3. By analyzing Table 3, We found that on dataset-1, differences in location interfered with CSI, resulting all network trained using CSI only that could not achieve the desired recognition results. In contrast, CM-NET uses soft labels generated by the vision-based network to supervise the training of the CSI network, allowing the CSI network to efficiently extract more useful human feature data and the network to achieve produce more accurate results.

**Table 3.** Performance of existing networks on dataset-1.

| Model | Accuracy | Recall | Precision | F1-Score |
|---|---|---|---|---|
| CM-NET | 86.40% | 85.44% | 85.68% | 85.15% |
| Two-Stream Transformer [24] | 72.24% | 70.92% | 64.77% | 66.50% |
| DNN [27] | 53.88% | 33.33% | 17.96% | 23.34% |
| SVM [26] | 49.41% | 41.71% | 42.25% | 40.17% |
| KNN [26] | 49.55% | 41.83% | 41.45% | 41.37% |

The five current networks' performance tests are shown in Table 4 for dataset-2. Table 4, through analysis, revealed that on dataset-2, the interference caused by the location is more apparent due to the position's immobility, causing all networks' accuracy to fall. Each network metric has a pronounced degradation when CSI is solely employed, especially for machine learning networks, where performance is approximately halved. In contrast, CM-NET increases the network's ability to extract features about the number of people by transferring the information learned from the vision network to the CSI network, reducing interference caused by location changes and ensuring stability for the network's performance.

**Table 4.** Performance of existing networks on dataset-2.

| Model | Accuracy | Recall | Precision | F1-Score |
|---|---|---|---|---|
| CM-NET | 83.61% | 83.40% | 82.92% | 82.65% |
| Two-Stream Transformer [24] | 53.51% | 53.11% | 37.00% | 42.65% |
| DNN [27] | 39.73% | 33.33% | 13.24% | 18.96% |
| SVM [26] | 25.92% | 24.32% | 30.31% | 22.28% |
| KNN [26] | 22.84% | 22.16% | 22.37% | 22.25% |

The results in Tables 3 and 4 reveals that in both cases—especially when the location is not fixed—CM-NET achieves superior people counting performance.

## 5. Conclusions

In this paper, we introduce multimodal knowledge distillation to the task of CSI-based people counting. We propose an end-to-end supervised cross-modal learning network, CM-NET. By transferring knowledge from a vision-based network to a CSI-based network using deep learning and distillation learning, CM-NET enhances the performance of CSI-based networks for people counting. Our data acquisition uses only a pair of receivers, transmitters, and a camera, and experiments are conducted in a real room. The experimental results show that the average recognition accuracy of CM-NET for up to 9 people is 86%, which is better than existing related methods. Our method is quite robust in complex indoor environments. For our future work, we plan to further improve the recognition accuracy of people counting.

**Author Contributions:** Conceptualization, Supervision, Project Administration, Writing—Original Draft, J.G.; Formal Analysis, Methodology, Software, Writing—Original Draft, X.G.; Visualization, Investigation, Z.L.; Writing—Review & Editing, Investigation, M.J.; Data Curation, Formal Analysis, J.W.; Visualization, Writing—Review & Editing, X.Y.; Conceptualization, Funding Acquisition, Resources, Supervision, Writing—Review & Editing, P.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key Research and Development Program of China grant number 2018YFB1802401.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

**Acknowledgments:** This work was partially supported by National Natural Science Foundation of China under grant agreements Nos. 61902316, 62133012, 61973250, 62073218, 61973249, 61902313, 62002271. Shaanxi Provincial Department of Education serves local scientific research under 19JC038, the Key Research and Development Program of Shaanxi under 2021GY-077, the popular development of deep integration of digital economy industry and green ecological industry: 150012100001, Young science and technology nova of Shaanxi Province: 2022KJXX-73 and the Fundamental Research Funds for the Central Universities under grant No. XJS210310.

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

## References

1. Wahl, F.; Milenkovic, M.; Amft, O. A distributed pir-based approach for estimating people count in office environments. In Proceedings of the 2012 IEEE 15th International Conference on Computational Science and Engineering, Paphos, Cyprus, 5–7 December 2012; pp. 640–647.
2. Shih, O.; Rowe, A. Occupancy estimation using ultrasonic chirps. In Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems, Seattle, WC, USA, 14–16 April 2015; pp. 149–158.
3. Rahman, H.; Han, H. Occupancy estimation based on indoor $CO_2$ concentration: Comparison of neural network and bayesian methods. *Int. J. Air-Cond. Refrig.* **2017**, *25*, 1750021. [CrossRef]
4. 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, 20–25 June 2005; Volume 1, pp. 886–893.
5. Yang, Y.; Wang, J.; Yang, Y. Improving svm classifier with prior knowledge in microcalcification detection1. In Proceedings of the 2012 19th IEEE International Conference on Image Processing, Orlando, FL, USA, 30 September–3 October 2012; pp. 2837–2840.
6. Nakatsuka, M.; Iwatani, H.; Katto, J. A study on passive crowd density estimation using wireless sensors. In Proceedings of the 4th International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2008), Tokyo, Japan, 11–13 June 2008.
7. Yoshida, T.; Taniguchi, Y. Estimating the number of people using existing wifi access point in indoor environment. In Proceedings of the 6th European Conference Computer Science, Rome, Italy, 7–9 November 2015; pp. 46–53.
8. Halperin, D.; Hu, W.; Sheth, A.; Wetherall, D. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM Comput. Commun. Rev.* **2011**, *41*, 53. [CrossRef]
9. Xi, W.; Zhao, J.; Li, Xi.; Zhao, K.; Tang, S.; Liu, X.; Jiang, Z. Electronic frog eye: Counting crowd using wifi. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 361–369.
10. Niu, R.; Blum, R.S.; Varshney, P.K.; Drozd, A.L. Target localization and tracking in noncoherent multiple-input multiple-output radar systems. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 1466–1489. [CrossRef]
11. Choi, J.W.; Yim, D.H.; Cho, S.H. People counting based on an ir-uwb radar sensor. *IEEE Sens. J.* **2017**, *17*, 5717–5727. [CrossRef]
12. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
13. Wu, B.; Nevatia, R. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *Int. J. Comput. Vis.* 2007, 75, 247–266.
14. Chen, L.; Tao, J.; Tan, Y.-P.; Chan, K.-L. People counting using iterative mean-shift fitting with symmetry measure. In Proceedings of the 2007 6th International Conference on Information, Communications & Signal Processing, Lisbon, Portugal, 28–30 June 2007; pp. 1–4.
15. Davies, A.C.; Yin, J.H.; Velastin, S.A. Crowd monitoring using image processing. *Electron. Commun. Eng. J.* **1995**, *7*, 37–47. [CrossRef]
16. Kong, D.; Gray, D.; Tao, H. A viewpoint invariant approach for crowd counting. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; Volume 3, pp. 1187–1190.
17. Hassan, M.A.; Pardiansyah, I.; Malik, A.S.; Faye, I.; Rasheed, W. Enhanced people counting system based head-shoulder detection in dense crowd scenario. In Proceedings of the 2016 6th International Conference on Intelligent and Advanced Systems (ICIAS), Kuala Lumpur, Malaysia, 15–17 August 2016; pp. 1–6.
18. Marana, A.N.; Velastin, S.A.; Costa, L.F.; Lotufo, R.A. Estimation of crowd density using image processing. In Proceedings of the IEE Colloquium on Image Processing for Security Applications, London, UK, 10 March 1997.

19. Chan, A.B.; Liang, Zh.J.; Vasconcelos, N. Privacy preserving crowd monitoring: Counting people without people models or tracking. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, 23–28 June 2008; pp. 1–7.
20. Choi, J.W.; Kim, J.H.; Cho, S.H. A counting algorithm for multiple objects using an ir-uwb radar system. In Proceedings of the 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, Beijing, China, 21–23 September 2012; pp. 591–595.
21. Alsamhi, S.H.; Almalki, F.; Ma, O.; Ansari, M.S.; Lee, B. Predictive estimation of optimal signal strength from drones over IoT frameworks in smart cities. *IEEE Trans. Mob. Comput.* **2021**, *22*, 402–416. [CrossRef]
22. Domenico, S.D.; Sanctis, M.D.; Cianca, E.; Bianchi, G. A trained-once crowd counting method using differential wifi channel state information. In Proceedings of the 3rd International on Workshop on Physical Analytics, Singapore, 25–30 June 2016; pp. 37–42.
23. Alsamhi, S.H.; Rajput, N.S. An efficient channel reservation technique for improved qos for mobile communication deployment using high altitude platform. *Wirel. Pers. Commun.* **2016**, *91*, 1095–1108. [CrossRef]
24. Li, B.; Cui, W.; Wang, W.; Zhang, L.; Chen, Z.; Wu, M. Two-stream convolution augmented transformer for human activity recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 286–293.
25. Sheng, B.; Gui, L.; Xiao, F. Ts-net: Device-free action recognition with cross-modal learning. In Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications, Nanjing, China, 25–27 June 2021; pp. 404–415.
26. Ding, Y.; Guo, B.; Xin, T.; Wang, P.; Wang, Z.; Yu, Z. Wicount:A WiFi-CSI based people counting method. *Comput. Sci.* **2019**, *4*, 38–52. (In Chinese)
27. Zhao, Y.; Liu, S.; Xue, F.; Chen, B.; Chen, X. Deepcount: Crowd counting with wi-fi using deep learning. *J. Commun. Inf. Netw.* **2019**, *4*, 38–52. [CrossRef]

*electronics*

MDPI

*Article*

# An Improved Crystal Structure Algorithm for Engineering Optimization Problems

**Wentao Wang, Jun Tian * and Di Wu**

College of Software, Nankai University, Tianjin 300071, China
* Correspondence: jtian@nankai.edu.cn

**Abstract:** Crystal Structure Algorithm (CryStAl) is a new meta-heuristic algorithm, and it has been studied by many scholars because of its wide adaptability and the fact that there is no need to set parameters in advance. An improved crystal structure algorithm (GLCryStAl) based on golden sine operator and Levy flight is designed in this paper. The algorithm makes good use of the relationship between the golden sine operator and the unit circle to make the algorithm exploration space more comprehensive, and then gradually narrows the search space in the iterative process, which can effectively speed up the convergence rate of the algorithm. At the same time, a Levy operator is introduced to help the algorithm effectively get rid of the attraction of local optimal value. To evaluate the performance of GLCryStAl, 12 classic benchmark functions and eight CEC2017 test functions were selected to design a series of comparative experiments. In addition, the experimental data of these algorithms are analyzed using the Wilcoxon and Friedman tests. Through these two tests, it can be found that GLCryStAl has significant advantages over other algorithms. Finally, this paper further tests the optimization performance of GLCryStAl in engineering design. GLCryStAl was applied to optimize pressure vessel design problems and tension/compression spring design problems. The optimization results show that GLCryStAl is feasible and effective in optimizing engineering design.

**Keywords:** crystal structure algorithm; golden sine algorithm; levy flight; engineering optimization problems

**Citation:** Wang, W.; Tian, J.; Wu, D. An Improved Crystal Structure Algorithm for Engineering Optimization Problems. *Electronics* **2022**, *11*, 4109. https://doi.org/10.3390/electronics11244109

Academic Editor: Ping-Feng Pai

Received: 11 November 2022
Accepted: 8 December 2022
Published: 9 December 2022

## 1. Introduction

Nowadays, practical problems in the fields of traffic scheduling, engineering design, machinery manufacturing, etc., are becoming more and more complex and challenging. When dealing with these problems, people often use optimization algorithms to save some costs. Since most production practice problems are multivariate, nonlinear, and have many complex constraints, the traditional conjugate gradient method does not perform well in the optimization of these problems [1]. The meta-heuristic algorithm has the characteristics of not depending on gradient information and wide adaptability, which can effectively make up for the shortcomings of traditional optimization algorithms. At present, the meta-heuristic algorithm is applied in various industries, such as workshop scheduling [2], task planning [3], engineering management [4–7] and so on.

The meta-heuristic algorithm is a mathematical method inspired by the biological behavior and some physical phenomena in nature [8]. The meta-heuristic algorithms mainly include the swarm optimization algorithm [9], evolutionary algorithm [10], physical and chemical algorithms [11] and human based algorithms [12]. The simulated annealing algorithm [13,14] and differential evolution algorithm [15] are classical evolutionary algorithms. Some classic and newly proposed swarm intelligence optimization algorithms are as follows: the bee collecting pollen algorithm (BCPA) [16] is proposed by simulating the behavior of bees collecting pollen. The fruit fly optimization algorithm (FOA) [17] is proposed by simulating the process of drosophila predation using its keen sense of smell and vision. The bat-inspired algorithm (BA) [18] is proposed by simulating bats using a sonar to detect prey and avoid obstacles. The grey wolf optimizer (GWO) [19] and the tuna swarm optimization (TSO) [20] are proposed by simulating the hunting behavior of wolves and tuna swarms. At present, the metaheuristic algorithm is applied in more and more fields, which has caused more and more scholars to study it.

In nature, a large number of microscopic material units such as molecules, atoms and ions are regularly arranged to form material structures called crystals. "Lattice points" are the basic units of crystal; they form a periodic lattice in a predefined space. The "basis" associated with each lattice point determines the position of the molecule in the crystal structure. In 2021, SIAMAK TALATAHAR proposed a new meta-heuristic algorithm called crystal structure algorithm (CryStAl) [21], by studying the principle of adding basis to form crystal structures with lattice points. The crystal structure algorithm has the advantages of simple structure, no need to set parameters in advance and strong adaptability, so it has generated widespread interest since it was proposed. Although CryStAl has excellent performance in many aspects, there are still some deficiencies in the crystal structure algorithm. CryStAl is susceptible to local extremes during iteration, resulting in insufficient exploration. Furthermore, the convergence speed of CryStAl is relatively slow. Up to now, no scholars have made relevant improvements to the crystal structure algorithm. Because the crystal structure algorithm has the advantages of simple structure, being easy to understand and low time complexity, this paper improves the algorithm under the premise of retaining these advantages of the crystal structure algorithm so that it can obtain better comprehensive optimization performance.

In view of these shortcomings, this paper developed a crystal structure algorithm improved by using a Levy flight operator and golden sine operator.

To ensure CryStAl has a balanced exploration and development performance, random numbers subject to Levy distribution are added to the CryStAl, which can effectively avoid CryStAl being affected by the suboptimal solution iterative process, and the golden sine operator is applied to modify the update strategy of candidate solutions, so that the algorithm converges faster. GLCryStAl is designed according to the modification strategy above.

The main contents of the article are summarized as follows. Section 2 introduces the CryStAl algorithm. Section 3 introduces the improved algorithm GLCryStAl in detail. In Section 4, a series of comparative experiments are designed and experimental numerical analysis is carried out. Section 5 uses GLCryStAl to optimize two engineering problems. Section 6 provides a comprehensive discussion of GLCryStAl. The paper ends in Section 7.

## 2. An Overview of the Crystal Structure Algorithm

The crystal structure algorithm is inspired by the principle of adding basis to lattice points to form crystals. Based on this principle, SIAMAK TALATAHAR proposed a crystal structure algorithm in 2021.

The internal particles of the crystal are regularly arranged. The structural particles that make up a crystal are regularly arranged at certain points in three-dimensional space, and these points periodically form an infinite lattice with a certain geometric shape, called a "lattice". The "basis" associated with each lattice point in the crystal determines the position of the particle in the crystal structure, adding basis to the lattice point to form the crystal structure. The mathematical model of the lattice position is as follows [22]:

$$r = \sum n_i a_i \tag{1}$$

where $i$ represents the number of crystal angles, $a_i$ is the shortest vector and $n_i$ is an integer.

CryStAl initializes crystals using the following formula:

$$Cr = \begin{bmatrix} Cr_1 \\ Cr_2 \\ \vdots \\ Cr_i \\ \vdots \\ Cr_n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^j & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^j & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i^1 & x_i^2 & \cdots & x_i^j & \cdots & x_i^d \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^j & \cdots & x_n^d \end{bmatrix}, \begin{cases} i = 1, 2, \cdots, n \\ j = 1, 2, \cdots, d \end{cases} \tag{2}$$

where $n$ is the total number of single crystals and $d$ is the total number of variables. Single crystals are initialized using the following formula:

$$x_i^j(0) = x_{i,\min}^j + \xi(x_{i,\max}^j - x_{i,\min}^j), \begin{cases} i = 1, 2, \cdots, n \\ j = 1, 2, \cdots, d \end{cases} \tag{3}$$

where $x_{i,\max}^j$ and $x_{i,\min}^j$ denote the two extreme values of the $j^{th}$ decision variable of the $i^{th}$ candidate solution, $\xi$ is a random value in the range of 0–1 and $x_i^j(0)$ denotes the initial position of the single crystal.

In CryStAl, the crystal at the corner is called $Cr_{main}$. The average value of randomly selected crystals is $Fc$. Currently, the best crystal is $Cr_b$. The crystal structure algorithm has four candidate solution updating strategies, as follows:

Simple cubicle strategy:

$$Cr_{new} = Cr_{old} + rCr_{main} \tag{4}$$

Cubicle with the optimal crystal:

$$Cr_{new} = Cr_{old} + r_1Cr_{main} + r_2Cr_b \tag{5}$$

Cubicle with the average crystal:

$$Cr_{new} = Cr_{old} + r_1Cr_{main} + r_2Fc \tag{6}$$

Cubicle with the optimal crystal and average crystal:

$$Cr_{new} = Cr_{old} + r_1Cr_{main} + r_2Cr_b + r_3Fc \tag{7}$$

where $r$–$r_3$ are four random values, $Cr_{old}$ is the position of the old crystal and $Cr_{new}$ is the position of the new crystal. The pseudocode of CryStAl is displayed in Algorithm 1:

**Algorithm 1. Pseudo-code of CryStAl**

```
Initialization: the positions of crystals, $Cr_i$ (I = 1, 2, . . . , $n$)
Calculate the fitness value of all crystals
while $t < t_{max}$ do
    for (each crystal) do
            Create $Cr_{main}$
            Create new crystals through formula (4)
            Create $Cr_b$
            Create new crystals through formula (5)
            Create $Fc$
           Create new crystals through formula (6)
           Create new crystals through formula (7)
           if (the new crystal goes beyond the preset boundary) then
                Modify the position of the new crystal
           end if
           Calculate the fitness values of all new crystals
           Update the crystal with the optimal fitness value
     end for
     $t = t + 1$
 end while
 return the best crystal
```

## 3. The Proposed Algorithm

In this chapter, we first introduce the golden sine operator and Levy flight operator. Based on these two operators, the original crystal structure algorithm is modified. Then, a crystal structure algorithm based on Levy flight and golden sine operators is proposed, which is called GLCryStAl.

### *3.1. Golden Sine Position Update Strategy*

The golden sine algorithm [23] is called golden-SA for short. It is a new intelligent optimization algorithm proposed by Tanyildizi, inspired by the mathematical model of sine function. Golden-SA is widely studied because of its simple structure, fast convergence speed and strong stability. Golden-SA simulates the search space exploration process by using the sine function to scan the unit circle. It makes good use of the special relationship between the sine function and the unit circle and combines the golden section coefficients to search the algorithm space iteratively, and finally finds the optimal solution set.

The golden section coefficient is a concept proposed by the ancient Greek mathematician Eudoxus. It does not depend on gradient information and only needs to be iterated once per step, while its contraction steps are fixed per step. Scholars have found that the strategy of combining the traditional sine function with the golden section coefficient can help the algorithm quickly find the extreme value of the unimodal function. At the same time, the golden sine search strategy has good ergodicity, so it can effectively prevent the algorithm from being attracted by the local extreme value.

The mathematical description of the Golden-SA strategy is shown in Equation (8):

$$V_i^{t+1} = V_i^t |\sin(r_1)| - r_2 \sin(r_1) \left| x_1 B_i^t - x_2 V_i^t \right| \tag{8}$$

where $t$ represents the current number of iterations and $B_i^t$ represents the position of the best individual in the population in the $t$th iteration. $r_1$ and $r_2$ are random numbers in [0,2π] and [0,π], respectively; they determine the moving distance of the current individual in the next iteration and the direction of the current individual position update. $V_i^t$ and $V_i^{t+1}$ represent the positions of the $i$th individual in iteration $t$ and iteration $t + 1$, respectively. $x_1$ and $x_2$ are the golden section coefficients mentioned above, which can help the algorithm gradually narrow the search space and guide the ordinary individuals in the population toward the optimal individuals.

The mathematical expressions of $x_1$ and $x_2$ are as follows:

$$x_1 = a \cdot (1 - \tau) + b \cdot \tau \tag{9}$$

$$x_2 = a \cdot \tau + b \cdot (1 - \tau) \tag{10}$$

$$\tau = \frac{\sqrt{5} - 1}{2} \tag{11}$$

where $a$ and $b$ are the search interval, and the golden sine algorithm is segmented by the interval of the standard sine function. Since the period of the standard sine function is $2\pi$, according to the relationship between the standard sine and the unit circle, in order to enable the population to traverse the searched space in each dimension throughout the period, the values of $a$ and $b$ are usually $\pi$ and $-\pi$. $\tau$ is the golden ratio.

### *3.2. Levy Flight Position Update Strategy*

Levy operator [24] is a search strategy consistent with Levy distribution, and its step size is random, which makes Levy operator more suitable for exploration in a wider space than Brownian motion [25]. In the search process, Levy flight uses long distance step in low frequency and short distance step in high frequency, which can effectively avoid the algorithm being attracted by local extrema in the optimization process. Due to the high complexity of Levy distribution, researchers often use the Mantegna [26] algorithm to simulate Levy flight step size, which is defined as follows:

$$s = \frac{\mu}{|\nu|^{1/\beta}} \tag{12}$$

where $\mu$ and $\nu$ are defined as follows:

$$\mu \sim N\left(0, \sigma_\mu^2\right) \tag{13}$$

$$\nu \sim N\left(0, \sigma_\nu^2\right) \tag{14}$$

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left[\frac{(1+\beta)}{2}\right] \cdot \beta \cdot 2^{\frac{(1+\beta)}{2}}} \right\}, \ \sigma_\nu = 1 \tag{15}$$

where the value of $\beta$ is usually 1.5.

In order to show the global exploration capability of Levy flight more intuitively, this paper compares Levy flight with random walk strategy. The comparison results are presented in Figure 1.

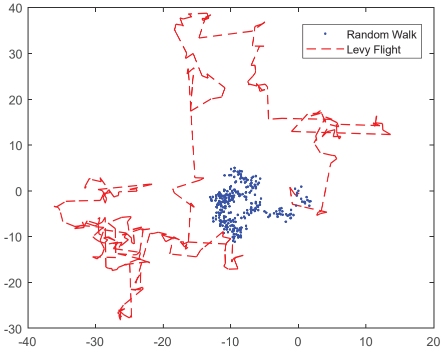


**Figure 1.** Simulation comparison experiment diagram of Levy flight and random walk.

Figure 1 shows that the Levy flight has a larger search range. The jump points of random walk strategy are more concentrated, and the jump points of Levy flight strategy are widely distributed. This means that the crystal structure algorithm modified by Levy flight can effectively enhance its global search performance.

### *3.3. Improved Crystal Structure Algorithm*

Aiming at the shortcomings of crystal structure algorithm, such as slow convergence speed and the requirement of practical engineering project for algorithm accuracy, this paper proposes a crystal structure algorithm (GLCryStAl) combining golden sine and Levy flight operator.

The GLCryStAl algorithm introduces the golden sine algorithm and the golden section coefficient into the three position update strategies mentioned in Equations (4)–(6), and determines its update position according to Equation (8). Because the golden sine operator has excellent ergodicity, it can make the optimization space of the algorithm more comprehensive and control the distance and direction of the candidate solution update through the parameters $r_1$ and $r_2$, so the exploration space of the algorithm can be gradually reduced. Therefore, the introduction of the golden sine operator can reduce the solution time of the algorithm and help the algorithm to obtain a more ideal solution.

Levy operator can significantly enhance the algorithm's global exploration performance. In this paper, the update strategy in Equation (7) is mutated by using the jump characteristics of the combination of long and short steps of the Levy operator, which can greatly improve the diversity of the algorithm population and make the optimization efficiency of the algorithm higher. The specific steps of GLCryStAl are as follows:

The modified simple cubicle strategy:

$$Cr_{new} = Cr_1|\sin(r_1)| - r_2\sin(r_1)|x_1 Cr_b - x_2 Cr_1| \tag{16}$$

The modified cubicle with the optimal crystal:

$$Cr_{new} = Cr_2|\sin(r_1)| - r_2\sin(r_1)|x_1 Cr_b - x_2 Cr_2| \tag{17}$$

The modified cubicle with the average crystal:

$$Cr_{new} = Cr_3|\sin(r_1)| - r_2\sin(r_1)|x_1 Cr_b - x_2 Cr_3| \tag{18}$$

Cubicle with the optimal crystal and average crystal:

$$Cr_{new} = Cr_{old} + \alpha \otimes Levy(\lambda) \otimes Cr_{main} + \alpha \otimes Levy(\lambda) \otimes Cr_b + \alpha \otimes Levy(\lambda) \otimes Fc \tag{19}$$

where $Cr_b$ is the optimal candidate in the current population. $Cr_1$, $Cr_2$ and $Cr_3$ are the candidate solutions produced by Equations (16)–(18) in the last execution,$\alpha$ is a distance control parameter, $\otimes$ denotes point multiplication, and $Levy(\lambda)$ is the jump path whose jump distance obeys Levy distribution. Considering that the exploration step size of Levy flight is too aggressive, it may jump out of the main search range during the algorithm search process, so this paper sets $\alpha$ to 0.01.

The pseudocode of GLCryStAl is displayed in Algorithm 2:

**Algorithm 2.** Pseudo-code of GLCryStAl

```
Initialization: the positions of crystals, $Cr_i$ (i = 1, 2, ... , $n$)
Calculate the fitness value of all crystals
while $t < t_{max}$ do
    for (each crystal) do
        Create $Cr_{main}$, $Cr_b$
        Create new crystals through formula (16)
        Create new crystals through formula (17)
        Create new crystals through formula (18)
        Create $Fc$
        Create new crystals through formula (19)
        if (the new crystal goes beyond the preset boundary) then
            Modify the position of the new crystal
        end if
        Calculate the fitness values of all new crystals
        Update the crystal with the optimal fitness value
    end for
    $t = t + 1$
end while
return the best crystal
```

## 4. Simulation Experiments and Results Analysis

In this section, 12 classic benchmark functions and eight CEC2017 test functions are applied to design comparative experiments of the other five algorithms of GLCryStAl in two different dimensions. To avoid the length of the article being too long, this paper selected part of the CEC2017 test function, which is also representative. Finally, the experimental results are numerically analyzed to verify the optimization performance of GLCryStAl.

### *4.1. Benchmark Functions and Experimental Design*

The details of the test functions are displayed in Table 1. $F_1$–$F_6$ are unimodal functions, which are applied to evaluate the solution speed of these algorithms. $F_7$–$F_9$ are multimodal functions, which are applied to evaluate whether these algorithms have excellent global exploration capacity. $F_{10}$–$F_{12}$ are combined functions, which are suitable for testing algorithm performance in fixed dimensions and are used to test the answer accuracy of these algorithms. $F_{13}$–$F_{20}$ are the CEC2017 functions which are applied to test the comprehensive capability of these algorithms.

Based on these 20 test functions, this paper designs a series of experiments comparing GLCryStAl with some of the latest algorithms and an improved algorithm. These competitor algorithms are CryStAl, Accelerated Particle Swarm Optimization Algorithm (APSO) [27], Whale Optimization Algorithm (WOA) [28], Golden Jackal Optimization (GJO) [29], Tunicate Swarm Algorithm (TSA) [30] and the newly proposed Dung beetle optimizer (DBO) [31]. Functions $F_1$–$F_9$ are tested in 30 and 100 dimensions, respectively, and $F_{10}$–$F_{12}$ are tested in their suitable dimension. Eight CEC2017 benchmark functions are tested in 50 dimensions. The maximum number of evaluations of $F_1$–$F_{12}$ is 1000. Since CEC2017 benchmark functions are too complex, the number of evaluations of eight CEC2017 functions are simplified to 3000 without losing representativeness. The population size of each algorithm is 30. To avoid accidental interference, we run each algorithm 30 times independently in each experiment. The parameter values of these algorithms involved in these experiments are shown in Table 2.

**Table 1.** Benchmark functions.

| Function | Dim | Range | $\mathbf{f_{min}}$ |
|---|---|---|---|
| $F_1(x) = \sum_{i=1}^{D} x_i^2$ | 30, 100 | [100, 100] | 0 |
| $F_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i-1}^{D} \lvert x_i \rvert$ | 30, 100 | $[-10, 10]$ | 0 |
| $F_3(x) = \sum_{i=1}^{D} \left(\sum_{j-1}^{D} x_i\right)^2$ | 30, 100 | $[-100, 100]$ | 0 |
| $F_4(x) = \max_i\{\lvert x_i \rvert, 1 \leq i \leq D\}$ | 30, 100 | $[-100, 100]$ | 0 |
| $F_5(x) = \sum_{i=1}^{D} 100\left(x_{i+1}^2 - x_i^2\right)^2 + (x_i - 1)^2$ | 30, 100 | $[-30, 30]$ | 0 |
| $F_6(x) = \sum_{i=1}^{D} i{x_i}^4 + \text{random}[0, 1)$ | 30, 100 | $[-1.28, 1.28]$ | 0 |
| $F_7(x) = \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$ | 30, 100 | $[-5.12, 5.12]$ | 0 |
| $F_8(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum\limits_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum\limits_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + \text{e}$ | 30, 100 | $[-32, 32]$ | $8.8818 \times 10^{-16}$ |
| $F_9(x) = \sum\limits_{i=1}^{D} \frac{x_i^2}{4000} - \prod\limits_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | 30, 100 | $[-600, 600]$ | 0 |
| $F_{10}(x) = \left((1/500) + \sum_{j=1}^{25} \left(1/\left(j + \sum_{i=1}^{2} (x_i - a_{ij})^6\right)\right)\right)^{-1}$ | 4 | $[-65.53, 65.53]$ | 0.998004 |
| $F_{11}(x) = \sum_{i=1}^{11} \left(a_i - \left(x_1(b_i^2 + b_i x_2)/b_i^2 + b_i x_3 + x_4\right)\right)^{-1}$ | 4 | $[-5, 5]$ | 0.0003075 |
| $F_{12}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right]$<br>$\times\left[30 + (2\,x_1 - 3x_2)^2(18 - 32\,x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$ | 2 | $[-5, 5]$ | $-1.03163$ |
| $F_{13}(x)(CEC2017\ 14: Hybrid\ Function\ 4\ (N = 4))$ | 50 | [100, 100] | 1400 |
| $F_{14}(x)(CEC2017\ 15: Hybrid\ Function\ 5\ (N = 4))$ | 50 | [100, 100] | 1500 |
| $F_{15}(x)(CEC2017\ 17: Hybrid\ Function\ 6\ (N = 5))$ | 50 | $[-100, 100]$ | 1700 |
| $F_{16}(x)(CEC2017\ 18: Hybrid\ Function\ 6\ (N = 5))$ | 50 | $[-100, 100]$ | 1800 |
| $F_{17}(x)(CEC2017\ 19: Hybrid\ Function\ 6\ (N = 5))$ | 50 | $[-100, 100]$ | 1900 |
| $F_{18}(x)(CEC2017\ 20: Hybrid\ Function\ 6\ (N = 6))$ | 50 | $[-100, 100]$ | 2000 |
| $F_{19}(x)(CEC2017\ 23: Composition\ Function\ 3\ (N = 4))$ | 50 | $[-100, 100]$ | 2300 |
| $F_{20}(x)(CEC2017\ 27: Composition\ Function\ 7\ (N = 6))$ | 50 | $[-100, 100]$ | 2700 |

**Table 2.** Parameter values of the algorithms.

| Algorithm | Parameter Value |
|---|---|
| APSO | $\alpha = 1,\ \beta = 0.5,\ \gamma = 0.95,\ population\ size\ \text{N} = 30,\ \text{t}_{\text{max}} = 1000, 3000$ |
| WOA | $population\ size\ \text{N} = 30,\ \text{t}_{\text{max}} = 1000, 3000$ |
| GJO | $population\ size\ \text{N} = 30,\ \text{t}_{\text{max}} = 1000, 3000$ |
| TSA | $P_{\text{min}} = 1, P_{\text{max}} = 4,\ population\ size\ \text{N} = 30,\ \text{t}_{\text{max}} = 1000, 3000$ |
| DBO | $k = 1, \lambda = 4,\ \text{b} = 0.3,\ S = 0.5,\ population\ size\ \text{N} = 30,\ \text{t}_{\text{max}} = 1000, 3000$ |
| CryStAl | $population\ size\ \text{N} = 30,\ \text{t}_{\text{max}} = 1000, 3000$ |
| GLCryStAl | $population\ size\ \text{N} = 30,\ \text{t}_{\text{max}} = 1000, 3000$ |

### *4.2. Results and Analysis*

Table 3 displays the experimental results of GLCryStAl and other competitors in low dimensional benchmark functions (dimension = 30). Std is the standard deviation and mean is the average. The mean reflects the solution accuracy of these algorithms and Std reflects their robustness. $F_{10}$–$F_{12}$ are tested in their suitable dimension.

Table 4 displays the test data of GLCryStAl and competitors in high-dimensional benchmark functions (dimension = 100). The experimental results of eight test functions in CEC2017 are shown in Table 5.

**Table 3.** Experimental results in 30 dimensions.

| Function | Performance | APSO | WOA | GJO | TSA | DBO | CryStAl | GLCryStAl |
|---|---|---|---|---|---|---|---|---|
| $F_1$ | Mean | $5.27 \times 10^{-39}$ | $4.31 \times 10^{-159}$ | $4.15 \times 10^{-113}$ | 0 | $7.46 \times 10^{-283}$ | $5.97 \times 10^{-32}$ | 0 |
| | Std | $1.16 \times 10^{-39}$ | $8.51 \times 10^{-159}$ | $8.45 \times 10^{-113}$ | 0 | 0 | $1.74 \times 10^{-31}$ | 0 |
| $F_2$ | Mean | $2.62 \times 10^{-1}$ | $1.45 \times 10^{-103}$ | $1.75 \times 10^{-65}$ | $3.75 \times 10^{-200}$ | $3.93 \times 10^{-122}$ | $4.58 \times 10^{-17}$ | 0 |
| | Std | $4.21 \times 10^{-1}$ | $3.74 \times 10^{-103}$ | $3.00 \times 10^{-65}$ | 0 | $1.24 \times 10^{-121}$ | $1.12 \times 10^{-16}$ | 0 |
| $F_3$ | Mean | 9.76 | $1.56 \times 10^{+4}$ | $1.90 \times 10^{-34}$ | 0 | $2.64 \times 10^{-185}$ | $6.03 \times 10^{-34}$ | 0 |
| | Std | 4.43 | $6.68 \times 10^{+3}$ | $5.71 \times 10^{-34}$ | 0 | 0 | $1.08 \times 10^{-33}$ | 0 |
| $F_4$ | Mean | $3.15 \times 10^{-1}$ | $4.77 \times 10^{+1}$ | $4.77 \times 10^{-34}$ | $8.15 \times 10^{-185}$ | $4.45 \times 10^{-107}$ | $7.54 \times 10^{-17}$ | 0 |
| | Std | $1.27 \times 10^{-1}$ | $2.98 \times 10^{+1}$ | $7.95 \times 10^{-34}$ | 0 | $1.41 \times 10^{-106}$ | $2.01 \times 10^{-16}$ | 0 |
| $F_5$ | Mean | $3.09 \times 10^{+1}$ | $2.74 \times 10^{+1}$ | $2.74 \times 10^{+1}$ | $2.89 \times 10^{+1}$ | $2.48 \times 10^{+1}$ | $2.87 \times 10^{+1}$ | $2.87 \times 10^{+1}$ |
| | Std | 6.28 | $5.90 \times 10^{-1}$ | $5.54 \times 10^{-1}$ | $3.20 \times 10^{-1}$ | $2.20 \times 10^{-1}$ | $2.66 \times 10^{-2}$ | $1.09 \times 10^{-1}$ |
| $F_6$ | Mean | $1.74 \times 10^{-1}$ | $6.67 \times 10^{-4}$ | $1.94 \times 10^{-4}$ | $7.17 \times 10^{-6}$ | $6.77 \times 10^{-4}$ | $5.37 \times 10^{-4}$ | $5.11 \times 10^{-5}$ |
| | Std | $6.11 \times 10^{-2}$ | $1.25 \times 10^{-3}$ | $8.02 \times 10^{-5}$ | $2.54 \times 10^{-5}$ | $3.26 \times 10^{-4}$ | $3.38 \times 10^{-4}$ | $3.39 \times 10^{-5}$ |
| $F_7$ | Mean | $8.51 \times 10^{+1}$ | 0 | 0 | 6.81 | 2.19 | 0 | 0 |
| | Std | $1.74 \times 10^{+1}$ | 0 | 0 | $1.84 \times 10^{+1}$ | 6.92 | 0 | 0 |
| $F_8$ | Mean | $3.43 \times 10^{-1}$ | $5.15 \times 10^{-15}$ | $4.44 \times 10^{-15}$ | $4.44 \times 10^{-15}$ | $8.88 \times 10^{-16}$ | $8.88 \times 10^{-16}$ | $8.88 \times 10^{-16}$ |
| | Std | $4.84 \times 10^{-1}$ | $2.13 \times 10^{-15}$ | 0 | 0 | 0 | 0 | 0 |
| $F_9$ | Mean | $5.57 \times 10^{-3}$ | 0 | 0 | $9.86 \times 10^{-4}$ | 0 | 0 | 0 |
| | Std | $1.33 \times 10^{-2}$ | 0 | 0 | $2.96 \times 10^{-3}$ | 0 | 0 | 0 |
| $F_{10}$ | Mean | $1.27 \times 10^{+1}$ | 2.37 | 5.10 | $1.13 \times 10^{+1}$ | 1.39 | $9.98 \times 10^{-1}$ | $9.98 \times 10^{-1}$ |
| | Std | $1.48 \times 10^{-13}$ | 2.90 | 4.64 | 6.25 | $8.37 \times 10^{-1}$ | 0 | 0 |
| $F_{11}$ | Mean | $5.04 \times 10^{-3}$ | $7.72 \times 10^{-4}$ | $4.96 \times 10^{-4}$ | $1.54 \times 10^{-2}$ | $7.02 \times 10^{-4}$ | $4.03 \times 10^{-4}$ | $3.09 \times 10^{-4}$ |
| | Std | $7.77 \times 10^{-3}$ | $5.34 \times 10^{-4}$ | $3.64 \times 10^{-4}$ | $2.61 \times 10^{-2}$ | $2.63 \times 10^{-4}$ | $5.58 \times 10^{-5}$ | $8.23 \times 10^{-7}$ |
| $F_{12}$ | Mean | 3.00 | 3.00 | 3.00 | $8.40 \times 10^{+1}$ | 5.70 | 3.00 | 3.00 |
| | Std | $5.63 \times 10^{-15}$ | $8.54 \times 10^{-6}$ | $2.37 \times 10^{-6}$ | $3.71 \times 10^{+1}$ | 8.54 | $5.85 \times 10^{-5}$ | $6.58 \times 10^{-6}$ |

**Table 4.** Experimental results in 100 dimensions.

| Function | Performance | APSO | WOA | GJO | TSA | DBO | CryStAl | GLCryStAl |
|---|---|---|---|---|---|---|---|---|
| $F_1$ | Mean | $2.10 \times 10^{+1}$ | $1.58 \times 10^{-153}$ | $2.34 \times 10^{-60}$ | 0 | $1.56 \times 10^{-235}$ | $3.43 \times 10^{-32}$ | 0 |
| | Std | 5.73 | $2.58 \times 10^{-153}$ | $3.03 \times 10^{-60}$ | 0 | 0 | $9.64 \times 10^{-32}$ | 0 |
| $F_2$ | Mean | $4.23 \times 10^{+1}$ | $4.67 \times 10^{-102}$ | $2.68 \times 10^{-37}$ | $5.23 \times 10^{-184}$ | $2.13 \times 10^{-124}$ | $5.41 \times 10^{-17}$ | 0 |
| | Std | 5.80 | $1.34 \times 10^{-101}$ | $2.51 \times 10^{-37}$ | 0 | $5.17 \times 10^{-124}$ | $1.49 \times 10^{-16}$ | 0 |
| $F_3$ | Mean | $2.32 \times 10^{+2}$ | $9.68 \times 10^{+5}$ | $1.26 \times 10^{-10}$ | 0 | $3.30 \times 10^{-97}$ | $4.59 \times 10^{-36}$ | 0 |
| | Std | $4.17 \times 10^{+1}$ | $1.11 \times 10^{+5}$ | $3.11 \times 10^{-10}$ | 0 | $1.04 \times 10^{-96}$ | $1.16 \times 10^{-35}$ | 0 |
| $F_4$ | Mean | 2.29 | $5.70 \times 10^{+1}$ | 1.57 | $4.01 \times 10^{-170}$ | $4.17 \times 10^{-124}$ | $8.48 \times 10^{-18}$ | 0 |
| | Std | $1.80 \times 10^{-01}$ | $3.48 \times 10^{+1}$ | 4.72 | 0 | $1.16 \times 10^{-123}$ | $1.46 \times 10^{-17}$ | 0 |
| $F_5$ | Mean | $5.84 \times 10^{+3}$ | $9.82 \times 10^{+1}$ | $9.81 \times 10^{+1}$ | $9.85 \times 10^{+1}$ | $2.49 \times 10^{+1}$ | $9.87 \times 10^{+1}$ | $9.81 \times 10^{+1}$ |
| | Std | $1.67 \times 10^{+3}$ | $3.64 \times 10^{-1}$ | $5.98 \times 10^{-1}$ | $4.12 \times 10^{-1}$ | $2.71 \times 10^{-1}$ | $5.35 \times 10^{-2}$ | $7.03 \times 10^{-2}$ |
| $F_6$ | Mean | $5.21 \times 10^{+2}$ | $5.58 \times 10^{-4}$ | $4.71 \times 10^{-4}$ | $4.82 \times 10^{-5}$ | $7.68 \times 10^{-4}$ | $3.42 \times 10^{-4}$ | $1.34 \times 10^{-5}$ |
| | Std | $3.41 \times 10^{+2}$ | $1.19 \times 10^{-3}$ | $2.31 \times 10^{-4}$ | $1.56 \times 10^{-5}$ | $5.46 \times 10^{-4}$ | $2.87 \times 10^{-4}$ | $3.96 \times 10^{-5}$ |
| $F_7$ | Mean | $4.49 \times 10^{+2}$ | $1.14 \times 10^{-14}$ | 0 | $3.98 \times 10^{-1}$ | 0 | 0 | 0 |
| | Std | $6.35 \times 10^{+1}$ | $3.41 \times 10^{-14}$ | 0 | $4.87 \times 10^{-1}$ | 0 | 0 | 0 |
| $F_8$ | Mean | 3.70 | $4.09 \times 10^{-15}$ | $1.12 \times 10^{-14}$ | $5.15 \times 10^{-15}$ | $1.24 \times 10^{-15}$ | $1.24 \times 10^{-15}$ | $8.88 \times 10^{-16}$ |
| | Std | $2.98 \times 10^{-1}$ | $2.49 \times 10^{-15}$ | $2.95 \times 10^{-15}$ | $1.42 \times 10^{-15}$ | $1.12 \times 10^{-15}$ | $1.07 \times 10^{-15}$ | 0 |
| $F_9$ | Mean | $4.67 \times 10^{-1}$ | 0 | 0 | $9.86 \times 10^{-4}$ | 0 | 0 | 0 |
| | Std | $8.78 \times 10^{-2}$ | 0 | 0 | $2.96 \times 10^{-3}$ | 0 | 0 | 0 |

**Table 5.** Test results of CEC2017 functions.

| Function | Performance | APSO | WOA | GJO | TSA | DBO | CryStAl | GLCryStAl |
|---|---|---|---|---|---|---|---|---|
| $F_{13}$ | Mean | $9.25 \times 10^{+8}$ | $1.39 \times 10^{+6}$ | $2.06 \times 10^{+6}$ | $4.39 \times 10^{+7}$ | $7.70 \times 10^{+5}$ | $5.68 \times 10^{+4}$ | $2.51 \times 10^{+4}$ |
| | Std | $1.69 \times 10^{+1}$ | $1.73 \times 10^{+6}$ | $6.62 \times 10^{+5}$ | $1.20 \times 10^{+7}$ | $5.51 \times 10^{+5}$ | $3.24 \times 10^{+4}$ | $4.71 \times 10^{+4}$ |
| $F_{14}$ | Mean | $1.84 \times 10^{+10}$ | $5.67 \times 10^{+5}$ | $1.20 \times 10^{+9}$ | $7.32 \times 10^{+8}$ | $1.29 \times 10^{+7}$ | $5.81 \times 10^{+5}$ | $2.78 \times 10^{+5}$ |
| | Std | $2.72 \times 10^{+6}$ | $5.34 \times 10^{+5}$ | $5.02 \times 10^{+8}$ | $2.09 \times 10^{+9}$ | $4.05 \times 10^{+7}$ | $3.01 \times 10^{+5}$ | $1.72 \times 10^{+5}$ |
| $F_{15}$ | Mean | $3.91 \times 10^{+4}$ | $4.27 \times 10^{+3}$ | $3.50 \times 10^{+3}$ | $5.76 \times 10^{+3}$ | $4.08 \times 10^{+3}$ | $3.20 \times 10^{+3}$ | $2.84 \times 10^{+3}$ |
| | Std | $3.81 \times 10^{+2}$ | $4.90 \times 10^{+2}$ | $4.56 \times 10^{+2}$ | $1.76 \times 10^{+3}$ | $4.96 \times 10^{+2}$ | $1.76 \times 10^{+2}$ | $2.29 \times 10^{+2}$ |
| $F_{16}$ | Mean | $7.24 \times 10^{+8}$ | $1.28 \times 10^{+7}$ | $1.79 \times 10^{+7}$ | $5.85 \times 10^{+7}$ | $8.01 \times 10^{+6}$ | $1.92 \times 10^{+5}$ | $1.28 \times 10^{+5}$ |
| | Std | $2.90 \times 10^{+6}$ | $5.98 \times 10^{+6}$ | $2.15 \times 10^{+7}$ | $6.03 \times 10^{+7}$ | $7.21 \times 10^{+6}$ | $7.11 \times 10^{+4}$ | $2.10 \times 10^{+5}$ |
| $F_{17}$ | Mean | $1.09 \times 10^{10}$ | $3.05 \times 10^{+6}$ | $2.43 \times 10^{+8}$ | $1.17 \times 10^{+9}$ | $1.63 \times 10^{+6}$ | $7.76 \times 10^{+5}$ | $2.35 \times 10^{+5}$ |
| | Std | $1.27 \times 10^{+6}$ | $4.00 \times 10^{+6}$ | $3.00 \times 10^{+8}$ | $1.43 \times 10^{+9}$ | $1.69 \times 10^{+6}$ | $4.57 \times 10^{+5}$ | $4.97 \times 10^{+5}$ |
| $F_{18}$ | Mean | $3.96 \times 10^{+3}$ | $4.34 \times 10^{+3}$ | $3.37 \times 10^{+3}$ | $3.96 \times 10^{+3}$ | $3.68 \times 10^{+3}$ | $3.23 \times 10^{+3}$ | $3.05 \times 10^{+3}$ |
| | Std | $1.04 \times 10^{+2}$ | $2.88 \times 10^{+2}$ | $5.28 \times 10^{+2}$ | $3.70 \times 10^{+2}$ | $1.90 \times 10^{+2}$ | $1.97 \times 10^{+2}$ | $1.16 \times 10^{+2}$ |
| $F_{19}$ | Mean | $6.11 \times 10^{+3}$ | $3.54 \times 10^{+3}$ | $3.34 \times 10^{+3}$ | $4.15 \times 10^{+3}$ | $3.39 \times 10^{+3}$ | $3.83 \times 10^{+3}$ | $3.34 \times 10^{+3}$ |
| | Std | $1.06 \times 10^{+2}$ | $2.09 \times 10^{+2}$ | $1.20 \times 10^{+2}$ | $1.50 \times 10^{+2}$ | $1.54 \times 10^{+2}$ | $1.24 \times 10^{+2}$ | $1.54 \times 10^{+2}$ |
| $F_{20}$ | Mean | $1.32 \times 10^{+4}$ | $4.76 \times 10^{+3}$ | $3.93 \times 10^{+3}$ | $4.84 \times 10^{+3}$ | $3.87 \times 10^{+3}$ | $4.84 \times 10^{+3}$ | $3.79 \times 10^{+3}$ |
| | Std | $1.75 \times 10^{+2}$ | $4.01 \times 10^{+2}$ | $1.44 \times 10^{+2}$ | $4.10 \times 10^{+2}$ | $1.68 \times 10^{+2}$ | $2.78 \times 10^{+2}$ | $3.63 \times 10^{+2}$ |

As can be seen from the above three tables, when the dimension is 30, GLCryStAl can obtain the theoretical optimal values in test functions $F_1$–$F_4$ and $F_7$–$F_9$; GLCryStAl still shows excellent performance in other functions. The solution accuracy of GLCryStAl in $F_5$ is slightly worse than that of WOA, GJO and DBO, and the solution accuracy of GLCryStAl in $F_6$ is not good, but the gap between GLCryStAl and other competitive algorithms in these two test functions is not large. The results are analyzed according to the Std value. Among the functions other than $F_5$ and $F_6$, the Std value of GLCryStAl is the smallest, which indicates that the optimization effect of GLCryStAl is very stable.

When the dimension is 100, GLCryStAl has the best solution accuracy in all benchmark functions, and GLCryStAl can find the theoretical optimal value in all functions except $F_5$ and $F_6$. All the data were analyzed according to Std value. The Std value of CL in $F_5$ was only ever worse than CryStAl, and the Std value of GLCryStAl in $F_6$ was the second best, indicating that GLCryStAl still has excellent robustness when solving high-dimensional problems. From the data in Table 5, we can see that the optimization performance of GLCryStAl does not decrease as the test case dimensions increase.

The experimental results of CEC2017 function indicate that all algorithms do not obtain the theoretical optimal value. However, GLCryStAl can achieve better optimization accuracy than other competitors in all test functions. This indicates that the GLCryStAl algorithm modified by Levy operator and golden sine operator can get rid of the attraction of local extreme value more efficiently when solving difficult optimization problems. At the same time, the development and exploration ability of the algorithm are balanced to ensure that the algorithm has a fast convergence rate.

Based on the above three tables for overall analysis, GLCryStAl has the characteristics of high accuracy in 85% of the function optimization problems in the classical benchmark function. Combined with the experimental results of two different dimensions, the Std value of GLCryStAl shows obvious advantages in 76% of the functions. It is not difficult to see that the optimization effect of GLCryStAl is not easily affected by contingency probability, and GLCryStAl can always maintain a stable solution accuracy. In the CEC2017 experimental environment, GLCryStAl can calculate more accurate results within a limited number of executions. None of the comparison algorithms calculate the theoretical optimal value in eight CEC2017 experiments, but this can illustrate that the proposed Levy operator and golden sine operator can accelerate the convergence speed of GLCryStAl and enable it to obtain a better global exploration vision.

Because the golden sine operator has excellent ergodicity, the optimization space of the algorithm can be more comprehensive. The introduction of the golden section coefficient in the iterative process further reduces the search space of the algorithm, which greatly improves the convergence rate of GLCryStAl. Using the Levy flight strategy to modify the candidate solution update formula expands the exploration scope of GLCryStAl, and reduces the probability that GLCryStAl is attracted by local optimal solution. The above data strongly prove the superiority of GLCryStAl and the effectiveness of the proposed operator.

Figure 2 shows the convergence curves of GLCryStAl and other six comparison algorithms on 20 test functions, where Figure 2a–i are the curves of $F_1$–$F_9$ in 100 dimensions, Figure 2j–l are the curves of $F_{10}$–$F_{12}$ in their suitable dimension and Figure 2m–u are the curves of eight CEC2017 functions.

The convergence curves of these algorithms indicate that GLCryStAl has more excellent convergence performance than its competitors. For simple optimization problems, GLCryStAl can obtain theoretical optimal values within 400–700 iterations. For complex and challenging problems, GLCryStAl can also maintain a faster convergence rate and get rid of the influence of local attraction points, and ultimately achieve higher optimization accuracy.

In order to further test whether GLCryStAl has an obvious enhancement advantage compared with other algorithms, this paper uses the Wilcoxon [32] statistical method and Friedman method to analyze the experimental data of these algorithms in 100-dimensional benchmark functions. The data of $F_{10}$–$F_{12}$ are based on their respective dimensions. The experimental data of eight CEC2017 benchmark functions are measured in 50 dimensions. The results of Friedman and Wilcoxon tests are listed in Tables 6 and 7.

The smaller the rank mean of the algorithm, the better its performance. As can be seen from Table 6, GLCryStAl has the smallest rank mean, CryStAl ranks second, followed by DBO, GJO, WOA, TSA, APSO. According to the statistical analysis results, the rank mean of the second-ranked competitive algorithm is almost twice that of GLCryStAl, and the rank mean of APSO is more than three times that of GL, which indicates that the improvement effect of the operator proposed in this paper is significant.

In the Wilcoxon statistical test results, if the $p$-value is less than 0.05 and close to 0, it proves that the experimental results of the two algorithms are significantly different. If the $p$-value exceeds 0.05, it proves that the experimental results of the two algorithms are not significantly different. If the $p$-value is equal to NaN, it proves that the experimental results of the two algorithms are not different.

As can be seen from Table 7, the $p$-values of GLCryStAl are basically less than 0.05 and close to 0, which means that GLCryStAl has significant advantages compared with other algorithms. A small number of the data in Table 7 are greater than 0.05, as are a small number for NaN. This is because the final solution accuracy of these competitive algorithms is not much different from that of GLCryStAl, but it can be seen from the convergence curves of all algorithms that although the solution accuracy of these competitive algorithms is not much different from that of GLCryStAl, their convergence speed is generally much slower than that of GLCryStAl.

**Table 6.** Friedman analysis results.

| Algorithm | Rank Mean |
|---|---|
| GLCryStAl | 1.70 |
| CryStAl | 3.35 |
| DBO | 3.43 |
| GJO | 3.75 |
| WOA | 4.48 |
| TSA | 4.73 |
| APSO | 6.58 |

**Table 7.** $p$-value of Wilcoxon statistical test results.

| Function | GLCryStAl vs. APSO | GLCryStAl vs. WOA | GLCryStAl vs. GJO | GLCryStAl vs. TSA | GLCryStAl vs. DBO | GLCryStAl vs. CryStAl |
|---|---|---|---|---|---|---|
| $F_1$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ | NaN | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ |
| $F_2$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ |
| $F_3$ | $1.73 \times 10^{-4}$ | $1.73 \times 10^{-4}$ | $1.73 \times 10^{-4}$ | NaN | $6.39 \times 10^{-5}$ | $1.73 \times 10^{-4}$ |
| $F_4$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ | $6.39 \times 10^{-5}$ |
| $F_5$ | $1.83 \times 10^{-4}$ | $7.69 \times 10^{-4}$ | $2.57 \times 10^{-2}$ | $2.83 \times 10^{-3}$ | $1.83 \times 10^{-4}$ | $1.83 \times 10^{-4}$ |
| $F_6$ | $1.83 \times 10^{-4}$ | $1.83 \times 10^{-4}$ | $2.83 \times 10^{-3}$ | $1.73 \times 10^{-2}$ | $1.71 \times 10^{-3}$ | $1.83 \times 10^{-4}$ |
| $F_7$ | $6.39 \times 10^{-5}$ | $3.68 \times 10^{-1}$ | NaN | $3.50 \times 10^{-2}$ | NaN | NaN |
| $F_8$ | $6.39 \times 10^{-5}$ | $1.98 \times 10^{-3}$ | $3.29 \times 10^{-5}$ | $3.29 \times 10^{-5}$ | $3.68 \times 10^{-1}$ | $3.68 \times 10^{-1}$ |
| $F_9$ | $6.39 \times 10^{-5}$ | NaN | NaN | $3.68 \times 10^{-1}$ | NaN | NaN |
| $F_{10}$ | $1.78 \times 10^{-4}$ | $9.10 \times 10^{-1}$ | $4.52 \times 10^{-2}$ | $1.83 \times 10^{-4}$ | $2.12 \times 10^{-2}$ | $3.45 \times 10^{-1}$ |
| $F_{11}$ | $1.83 \times 10^{-4}$ | $1.83 \times 10^{-4}$ | $9.70 \times 10^{-1}$ | $1.83 \times 10^{-4}$ | $2.57 \times 10^{-2}$ | $1.83 \times 10^{-4}$ |
| $F_{12}$ | $1.79 \times 10^{-4}$ | $6.78 \times 10^{-1}$ | $4.40 \times 10^{-4}$ | $7.91 \times 10^{-1}$ | $2.71 \times 10^{-3}$ | $3.85 \times 10^{-1}$ |
| $F_{13}$ | $1.83 \times 10^{-4}$ | $1.83 \times 10^{-4}$ | $1.83 \times 10^{-4}$ | $1.83 \times 10^{-4}$ | $4.40 \times 10^{-4}$ | $4.73 \times 10^{-1}$ |
| $F_{14}$ | $1.83 \times 10^{-4}$ | $5.71 \times 10^{-1}$ | $1.83 \times 10^{-4}$ | $1.83 \times 10^{-4}$ | $2.83 \times 10^{-3}$ | $5.21 \times 10^{-1}$ |
| $F_{15}$ | $1.83 \times 10^{-4}$ | $2.11 \times 10^{-2}$ | $7.57 \times 10^{-2}$ | $1.83 \times 10^{-4}$ | $5.80 \times 10^{-3}$ | $5.83 \times 10^{-4}$ |
| $F_{16}$ | $1.83 \times 10^{-4}$ | $1.83 \times 10^{-4}$ | $1.83 \times 10^{-4}$ | $1.83 \times 10^{-4}$ | $1.83 \times 10^{-4}$ | $2.41 \times 10^{-1}$ |
| $F_{17}$ | $1.83 \times 10^{-4}$ | $9.10 \times 10^{-1}$ | $5.39 \times 10^{-2}$ | $1.83 \times 10^{-4}$ | $9.70 \times 10^{-1}$ | $2.41 \times 10^{-1}$ |
| $F_{18}$ | $1.83 \times 10^{-4}$ | $1.01 \times 10^{-3}$ | $4.27 \times 10^{-1}$ | $4.40 \times 10^{-4}$ | $5.83 \times 10^{-4}$ | $2.73 \times 10^{-1}$ |
| $F_{19}$ | $1.83 \times 10^{-4}$ | $2.73 \times 10^{-1}$ | $3.30 \times 10^{-4}$ | $7.69 \times 10^{-4}$ | $6.40 \times 10^{-2}$ | $8.50 \times 10^{-1}$ |
| $F_{20}$ | $1.83 \times 10^{-4}$ | $1.04 \times 10^{-1}$ | $9.70 \times 10^{-1}$ | $9.11 \times 10^{-3}$ | $1.71 \times 10^{-3}$ | $1.04 \times 10^{-1}$ |

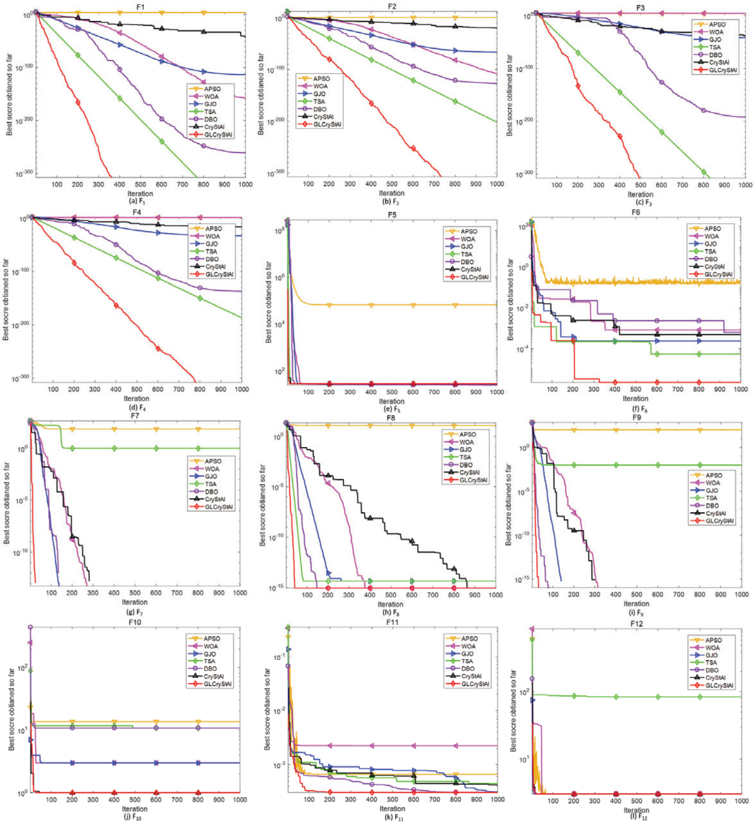


**Figure 2.** *Cont*.

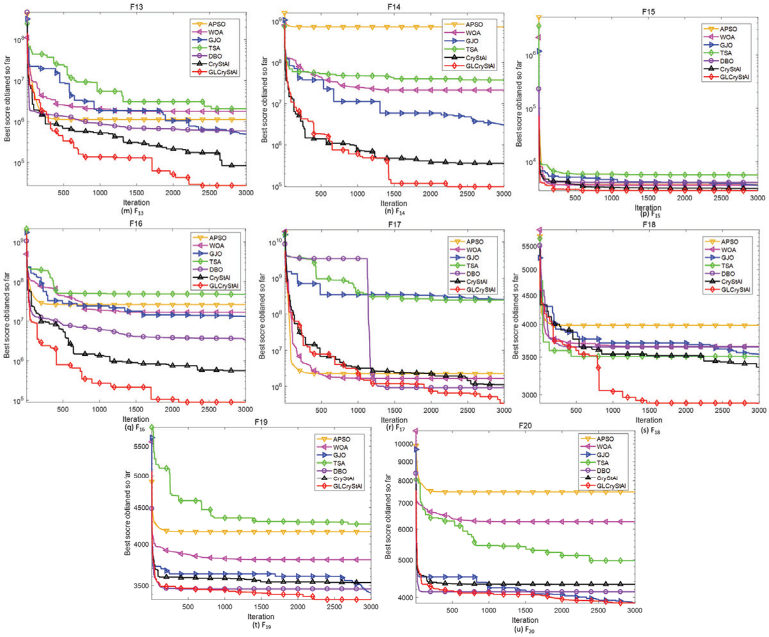


**Figure 2.** Convergence curve of each algorithm.

## 5. Optimization Engineering Example Using GLCryStAl

### *5.1. Using GLCryStAl to Optimize Pressure Vessel Design Problem*

The design of pressure vessel is a classical and important problem in practical engineering projects. The structure of the pressure vessel is shown in Figure 3, which consists of a cylindrical container and two hemispherical containers. In this optimization problem, there are four key variables, which are $T_s$, $T_h$, $R$ and $L$. $T_s$ denotes the thickness of the shell of the cylindrical container, $T_h$ denotes the thickness of the hemispheric container lid, $R$ is the radius inside the hemispherical container and $L$ is the length of the cylinder container. These four variables can be expressed as the following equation when using algorithms to optimize the pressure vessel design problem.

$$x = (x_1, x_2, x_3, x_4)^T = (T_s, T_h, R, L)^T \tag{20}$$

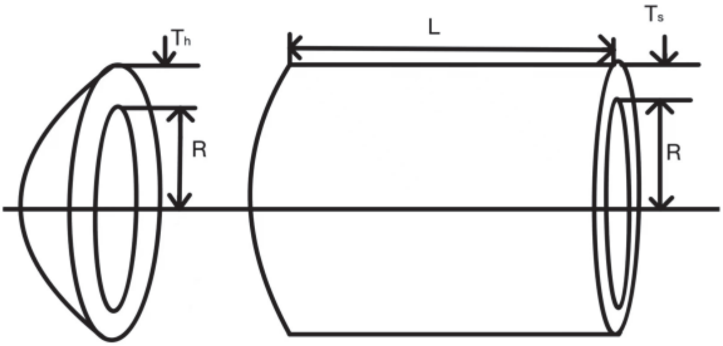


**Figure 3.** Pressure vessel structure diagram.

In this optimization problem, we use heuristic algorithm to regulate these four variables, so as to minimize the sum of material cost, forming cost and welding cost of the pressure vessel, which is transformed into the problem of finding the minimum of objective function with constraints.

The objective function of the pressure vessel design optimization problem is as follows:

$$\begin{aligned} \min f\left(\vec{x}\right) &= 0.6224x_1x_3x_4 + 1.7781x_2{x_3}^2 + 3.1661{x_1}^2x_4 + 19.84{x_1}^2x_3; \\ \text{s.t.}\quad & g_1(x) = -x_1 + 0.0193x_3 \le 0; \\ & g_2(x) = -x_2 + 0.00954x_3 \le 0; \\ & g_3(x) = -\pi {x_3}^2x_4 - \tfrac{4}{3}\pi x_3^3 + 1296000 \le 0; \\ & g_4(x) = x_4 - 240 \le 0; \\ & 1 \le x_1, x_2 \le 99, 10 \le x_3, x_4 \le 200 \end{aligned} \tag{21}$$

To verify the effectiveness of GLCryStAl in pressure vessel design, this paper selects PSO, WOA, GJO, TSA, DBO and CryStAl to compare with GLCryStAl, and runs each algorithm 10 times. The population size of all algorithms is 30, and the maximum number of iterations $t_{\max}$ = 1000. The specific results are listed in Tables 8 and 9.

**Table 8.** Total cost of six algorithms for optimizing pressure vessel design problems.

| **Algorithm** | **PSO** | **WOA** | **GJO** | **TSA** | **DBO** | **CryStAl** | **GLCryStAl** |
|---|---|---|---|---|---|---|---|
| Best value | 6233.24 | 7073.26 | 5935.81 | 7016.74 | 5949.13 | 5951.03 | 5912.19 |
| Worst value | 6401.17 | 9776.40 | 7051.61 | 7765.10 | 7319.00 | 6727.56 | 6745.60 |
| Mean | 6215.85 | 8345.98 | 6359.02 | 7529.88 | 6365.51 | 6175.29 | 6142.88 |
| Std | 112.39 | 853.42 | 537.95 | 269.94 | 618.57 | 241.23 | 204.76 |

**Table 9.** The best results of six algorithms.

| **Algorithm** | $T_s$ | $T_h$ | $R$ | $L$ | $f(x)$ |
|---|---|---|---|---|---|
| PSO | 0.9439015 | 0.466571 | 48.90681 | 107.2621 | 6233.24 |
| WOA | 0.8127619 | 0.7146932 | 40.51463 | 197.303 | 7073.26 |
| GJO | 0.7838487 | 0.3997165 | 40.60734 | 196.2115 | 5935.81 |
| TSA | 1.05297 | 0.554377 | 51.6783 | 86.8777 | 7016.74 |
| DBO | 0.81254 | 0.402322 | 42.098439 | 176.6367 | 5949.13 |
| CryStAl | 0.7923016 | 0.3923043 | 40.81446 | 193.5723 | 5951.03 |
| GLCryStAl | 0.7839314 | 0.3905878 | 40.6004 | 196.3812 | 5912.19 |

It is clear from the above two tables that the total cost calculated by GLCryStAl is the smallest in the pressure vessel optimization design problem. In addition, the Std of GLCryStAl is within an acceptable range, which proves from another aspect that GLCryStAl has excellent stability when solving constrained engineering design optimization problems.

### *5.2. Optimizing the Design of Tension/Compression Spring Using GLCryStAl*

To fully prove the feasibility and effectiveness of the GLCryStAl algorithm in engineering optimization problems, this paper applies the GLCryStAl algorithm to the optimization design problem of a tension/compression spring. A schematic of the spring is shown in Figure 4. This optimization problem is to select the most appropriate wire diameter (D), the average coil diameter (L) and the number of effective coils (P) under a series of specific constraints, so as to minimize the weight of the spring. In this paper, the three variables D, L and P are represented as $x_1$, $x_2$ and $x_3$. The mathematical modeling of the optimization problem in this section is as follows:

$$
\begin{array}{ll}
 & \min f\left(\vec{x}\right) = (x_3+2)x_2x_1^2; \\
\text{s.t.} & g_1(x) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \le 0; \\
 & g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0; \\
 & g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0; \\
 & g_4(x) = \frac{x_1+x_2}{1.5} - 1 \le 0; \\
 & 0.05 \le x_1 \le 2, 0.25 \le x_2 \le 1.3, 2 \le x_3 \le 15
\end{array}
\quad (22)
$$

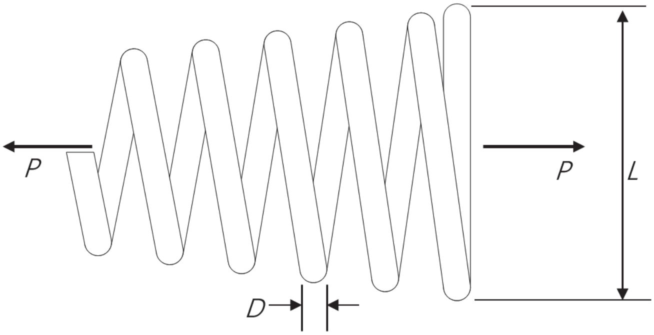


**Figure 4.** Schematic diagram of tension/compression spring design.

To evaluate the effectiveness of GLCryStAl in the design of tension/compression spring, a comparative experiment is designed. PSO, WOA, GJO, TSA, DBO and CryStAl are selected to compare with GLCryStAl, and the results are listed in Tables 10 and 11. In this comparison experiment, each algorithm runs 10 times independently. The population size of each algorithm is 30, and the maximum number of iterations $t_{\max}$ = 1000.

**Table 10.** Comparison results of six algorithms in spring design problem.

| **Algorithm** | **PSO** | **WOA** | **GJO** | **TSA** | **DBO** | **CryStAl** | **GLCryStAl** |
|---|---|---|---|---|---|---|---|
| Best value | 0.012773 | 0.012757 | 0.01273 | 0.013104 | 0.012703 | 0.012702 | 0.012692 |
| Worst value | 0.014331 | 0.017183 | 0.012745 | 0.015482 | 0.017773 | 0.012836 | 0.012801 |
| Mean | 0.013323 | 0.013703 | 0.012736 | 0.013690 | 0.013633 | 0.012783 | 0.012767 |
| Std | $5.774 \times 10^{-4}$ | $1.381 \times 10^{-3}$ | $8.093 \times 10^{-6}$ | $7.922 \times 10^{-4}$ | $1.721 \times 10^{-3}$ | $5.047 \times 10^{-5}$ | $5.135 \times 10^{-5}$ |

**Table 11.** The best results of six algorithms.

| **Algorithm** | ***D*** | ***L*** | ***P*** | $f(x)$ |
|---|---|---|---|---|
| PSO | 0.054129 | 0.41825 | 8.4229 | 0.012773 |
| WOA | 0.068001 | 0.89079 | 2.1716 | 0.017183 |
| GJO | 0.0503732 | 0.325773 | 13.4042 | 0.01273 |
| TSA | 0.055371 | 0.45034 | 7.4903 | 0.013104 |
| DBO | 0.0506135 | 0.331388 | 12.9636 | 0.012703 |
| CryStAl | 0.0523392 | 0.372375 | 10.4521 | 0.012702 |
| GLCryStAl | 0.0515739 | 0.353728 | 11.4895 | 0.012692 |

From the experimental results, in the optimization results of all algorithms, the mean value obtained by GLCryStAl is second only to the mean value obtained by GJO. However, from another point of view, the best value of spring weight optimized by the GLCryStAl algorithm is smaller than that of all the competitors. This proves that the GLCryStAl algorithm proposed in this paper is feasible and effective in the optimization design problem of a tension/compression spring.

## 6. Discussion of GLCryStAl

CryStAl is a heuristic algorithm with a simple structure and no need to set hyperparameters. This paper uses Levy operator and golden sine operator to make targeted optimization based on some subsidies of the original CryStAl algorithm. Firstly, this paper uses the golden sine operator to optimize the candidate solution update equation in CryStAl. The golden sine operator has excellent ergodicity to speed up the convergence of the algorithm. Secondly, this paper uses the levy flight operator to perturb the candidate solution update method of the CryStAl algorithm. By using the method of combining the long and short steps of the operator for optimization, the algorithm is effectively prevented from being attracted by the local optimal value during execution. The excellent optimization capability of GLCryStAl is demonstrated by comparison with various competitive algorithms. A series of experiments have proved the effectiveness of GLCryStAl from multiple viewpoints, which strongly indicates that GLCryStAl has a wide range of engineering applications.

In a series of function test experiments, GLCryStAl has higher solution accuracy than other competitors in 85% of classical benchmark functions. GLCryStAl has a higher Std value than other competitors in 76% of classical functions, which proves that the accuracy of GLCryStAl is not easily affected by random factors. In the comparison of CEC2017 benchmark functions, although all algorithms cannot calculate the theoretical optimal value, GLCryStAl can calculate a more accurate solution in a limited number of executions. The comparison results of a series of functions are analyzed by statistical methods. GLCryStAl achieved the smallest rank mean in the Friedman analysis experiment, and the *p*-value in the Wilcoxon analysis experiment is basically less than 0.05 and close to 0. These statistical analyses prove that the two operators proposed in this paper have significant improvement effects on CryStAl. Applying GLCryStAl to the pressure vessel design problem and the tension/compression spring design problem, compared with other competitors, GLCryStAl can calculate the minimum cost pressure vessel design and the minimum weight stretch/compression spring design. It can be proven from a series of experiments in this paper that the optimization performance of GLCryStAl is significantly improved compared with the original CryStAl algorithm, but the performance of GLCryStAl in solving some multi-peak complex problems still needs to be improved, which is also a major research direction in our future.

## 7. Conclusions

Focusing on the problem of the crystal structure algorithm being easily attracted by local extremum and the solution accuracy not being high enough, GLCryStAl is proposed. In GLCryStAl, the golden sine operator and Levy operator are applied to modify the update strategy of the four candidate solutions in the crystal structure. This improvement can effectively prevent GLCryStAl from being attracted by local extreme values in the optimization process, and the optimization speed of GLCryStAl is significantly accelerated. In this paper, 10 classical benchmark functions and eight CEC2017 test functions are applied to design a series of comparison experiments with the latest algorithms and our improved algorithm. When solving low-dimensional functions, GLCryStAl can calculate the theoretical optimal value of the function in $F_1$–$F_4$ and $F_7$–$F_9$, and GLCryStAl can maintain the minimum standard deviation in functions other than $F_5$ and $F_6$. When solving high-dimensional functions, GLCryStAl obtains theoretical optimal values in seven functions. It can be concluded from the experimental results that the optimization capability of GLCryStAl is obviously stronger. In addition, the experimental data of the six algorithms were analyzed by Wilcoxon and Friedman method. The statistical results show that the rank mean of GL is only 1.7, which indicates that GLCryStAl algorithm is superior to other competitors.

Finally, this paper uses GLCryStAl to optimize two practical engineering problems. In the first engineering problem, GLCryStAl is applied to the optimal design of pressure vessels. GLCryStAl achieved the best performance in this experiment. In the second engineering problem, GLCryStAl is applied to optimize the stretch/compression spring design. In this experiment, the optimal spring weight designed by GLCryStAl is second only to that of GJO. These two design problems fully demonstrate the feasibility and effectiveness of GLCryStAl in optimizing practical engineering problems.

**Author Contributions:** Conceptualization, W.W. and J.T.; methodology, W.W.; software, D.W.; validation, W.W.; formal analysis, D.W.; investigation, W.W.; resources, W.W.; data curation, W.W.; writing—original draft preparation, W.W.; supervision, J.T.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

## References

1. Wu, G.; Pedrycz, W.; Suganthan, P.N.; Mallipeddi, R. Avariabe reduction strategy for evolutionary algorithms handling equality constraints. *Appl. Soft Comput.* **2015**, *37*, 774–786. [CrossRef]
2. Liu, W.; Dridi, M.; Fei, H.; El Hassani, A.H. Hybrid metaheuristics for solving a home health care routing and scheduling problem with time windows, synchronized visits and lunch breaks. *Expert Syst. Appl.* **2021**, *183*, 115307. [CrossRef]
3. Wang, X.; Zhao, H.; Han, T.; Zhou, H.; Li, C. A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem. *Appl. Soft Comput.* **2019**, *78*, 240–260. [CrossRef]
4. Wang, Y.-G. A maximum-likelihood method for estimating natural mortality and catchability coefficient from catch-and-effort data. *Mar. Freshw. Res.* **1999**, *50*, 307–311. [CrossRef]
5. Wu, J.; Ding, Z. Improved grey model by dragonfly algorithm for Chinese tourism demand forecasting. In Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Kitakyushu, Japan, 22–25 September 2020.
6. Wu, J.; Cui, Z.; Chen, Y.; Kong, D.; Wang, Y.-G. A new hybrid model to predict the electrical load in five states of Australia. *Energy* **2019**, *166*, 598–609. [CrossRef]
7. Wang, L.; Zhou, G.; Xu, Y.; Wang, S.; Liu, M. An effective artificial bee colony algorithm for the flexible job-shop scheduling problem. *Int. J. Adv. Manuf. Technol.* **2012**, *60*, 303–315. [CrossRef]
8. Webb, B. *Swarm Intelligence: From Natural to Artificial Systems*; IEEE Press: Piscataway, NJ, USA, 2002; Volume 14, pp. 163–164.
9. Kennedy, J. Swarm Intelligence. In *Nature-Inspired and Innovative Computing*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 187–219.
10. Ashlock, D. *Evolutionary Computation for Modeling and Optimization*; Springer: New York, NY, USA, 2006.
11. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2015**, *27*, 495–513. [CrossRef]
12. Satapathy, S.; Naik, A. Social group optimization (SGO): A new population evolutionary optimization technique. *Complex Intell. Syst.* **2016**, *2*, 173–203. [CrossRef]
13. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
14. Wang, L.; Lee, S.N.S.; Hing, W.Y. Solving channel assignment problems using local search methods and simulated annealing. In *Independent Component Analyses, Wavelets, Neural Networks, Biosystems, and Nanoengineering IX*; SPIE: Bellingham, WA, USA, 2011; Volume 8058, pp. 490–497.
15. Storn, R.; Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
16. Lu, X.; Zhou, Y. A novel global convergence algorithm: Bee collecting pollen algorithm. In Proceedings of the International Conference on Intelligent Computing, Berlin, Germany, 15–18 September 2008.
17. Pan, W.T. A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowl. Based Syst.* **2012**, *26*, 69–74. [CrossRef]
18. Yang, X.S. A New Metaheuristic Bat-Inspired Algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
19. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
20. Xie, L.; Han, T.; Zhou, H.; Zhang, Z.-R.; Han, B.; Tang, A. Tuna swarm optimization: A novel swarm-based metaheuristic algorithm for global optimization. *Comput. Intell. Neurosci.* **2021**, *2021*, 9210050. [CrossRef] [PubMed]
21. Talatahari, S.; Azizi, M.; Tolouei, M.; Talatahari, B.; Sareh, P. Crystal Structure Algorithm (CryStAl): A Metaheuristic Optimization Method. *IEEE Access* **2021**, *9*, 71244–71261. [CrossRef]

22. Averill, B.A.; Eldredge, P. *Chemistry: Principles, Patterns, and Applications*; Benjamin Cummings: San Francisco, CA, USA, 2021.
23. Tanyildizi, E.; Demir, G. Golden Sine Algorithm: A Novel Math-Inspired Algorithm. *Adv. Electr. Comput. Eng.* **2017**, *17*, 71–78. [CrossRef]
24. Viswanathan, G.M.; Afanasyev, V.; Buldyrev, S.V.; Havlin, S.; Da Luz, M.G.E.; Raposo, E.P.; Stanley, H.E. Levy fights search patterns of biological organisms. *Phys. A Stat. Mech. Appl.* **2001**, *295*, 85–88. [CrossRef]
25. Biagini, F.; Hu, Y.; Øksendal, B.; Zhang, T. Stochastic optimal control and applications. In *Stochastic Calculus for Fractional Brownian Motion and Applications*; Springer Science & Business Media: Berlin, Germany, 2008; pp. 207–238.
26. Mantegna, R.N. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes. *Phys. Rev. E* **1994**, *49*, 4677–4689. [CrossRef]
27. Reddy, B.R.; Uttara, K.M. Performance Analysis of Mimo Radar Waveform Using Accelerated Particle Swarm Optimization Algorithm. *Signal Image Process.* **2012**, *3*, 4. [CrossRef]
28. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
29. Chopra, N.; Ansari, M.M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Syst. Appl.* **2022**, *198*, 116924. [CrossRef]
30. Kaur, S.; Awasthi, L.K.; Sangal, A.; Dhiman, G. Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Eng. Appl. Artif. Intell.* **2020**, *90*, 103541. [CrossRef]
31. Xue, J.; Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomput.* **2022**, 1–32. [CrossRef]
32. Rosner, B.; Glynn, R.J.; Lee, M.T. Incorporation of Clustering Effects for the Wilcoxon Rank Sum Test: A Large-Sample Approach. *Biometrics* **2003**, *59*, 1089–1098. [CrossRef] [PubMed]

*electronics*

*Article*

# Lightweight Multi-Scale Dilated U-Net for Crop Disease Leaf Image Segmentation

**Cong Xu, Changqing Yu and Shanwen Zhang ***

School of Electronic Information, Xijing University, Xi'an 710123, China
* Correspondence: zhangshanwen@xijing.edu.cn

**Abstract:** Crop disease leaf image segmentation (CDLIS) is the premise of disease detection, disease type recognition and disease degree evaluation. Various convolutional neural networks (CNN) and their modified models have been provided for CDLIS, but their training time is very long. Aiming at the low segmentation accuracy of various diseased leaf images caused by different sizes, colors, shapes, blurred speckle edges and complex backgrounds of traditional U-Net, a lightweight multi-scale extended U-Net (LWMSDU-Net) is constructed for CDLIS. It is composed of encoding and decoding sub-networks. Encoding the sub-network adopts multi-scale extended convolution, the decoding sub-network adopts a deconvolution model, and the residual connection between the encoding module and the corresponding decoding module is employed to fuse the shallow features and deep features of the input image. Compared with the classical U-Net and multi-scale U-Net, the number of layers of LWMSDU-Net is decreased by 1 with a small number of the trainable parameters and less computational complexity, and the skip connection of U-Net is replaced by the residual path (Respath) to connect the encoder and decoder before concatenating. Experimental results on a crop disease leaf image dataset demonstrate that the proposed method can effectively segment crop disease leaf images with an accuracy of 92.17%.

**Keywords:** crop disease leaf image segmentation (CDLIS); U-Net; dilated convolution; lightweight multi-scale dilated U-Net (LWMSDU-Net)

**Citation:** Xu, C.; Yu, C.; Zhang, S. Lightweight Multi-Scale Dilated U-Net for Crop Disease Leaf Image Segmentation. *Electronics* **2022**, *11*, 3947. https://doi.org/10.3390/electronics11233947

Academic Editor: Juan M. Corchado

Received: 8 October 2022
Accepted: 24 November 2022
Published: 29 November 2022

## 1. Introduction

Plant diseases severely affect the quality and yields of crops. Early detection of crop diseases reduces economic losses and has a positive impact on crop quality [1,2]. Crop disease leaf image segmentation (CDLIS) is a key prerequisite for the automatic detection, early warning, diagnosis and recognition of leaf diseases [3,4]. However, CDLIS is an important and challenging topic due to the various colors, shapes, textures, sizes and backgrounds of crop disease leaf images, as shown in Figure 1 [5,6].

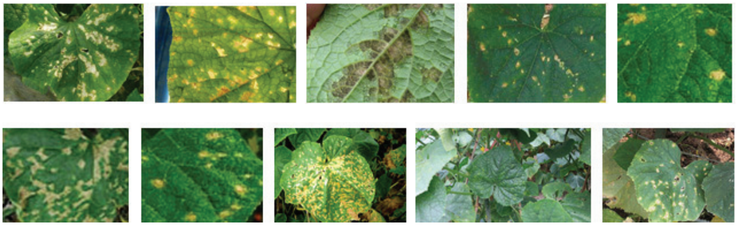


**Figure 1.** Disease leaf image examples.

Many image segmentation algorithms, such as fixed threshold, Otsu, K-means clustering, C-means clustering, fuzzy clustering, maximum entropy, 7 invariant moments and Local Binary Patterns (LBP), can be applied to CDLIS [7]. Wang et al. [8] proposed an adaptive CDLIS method based on K-means clustering by three stages. Fernandez et al. [9] applied principal component analysis (PCA) to the spectrum to evaluate the spectral separability between healthy and infected leaves, used the spectral ratio between infected and healthy leaves to determine the optimal wavelength for disease detection, and applied the linear support vector machine (SVM) classifier to some spectral features.

The accuracy of the above traditional algorithms mainly depends on experience, and due to the complexity of diseased leaf images, they lack generalization ability. With the improvement of computing power, storage, Internet of Things, big data and artificial intelligence, deep learning methods, such as convolutional neural network (CNN), full convolutional neural network (FCN) and U-Net, have been widely applied to the detection, segmentation and classification of crop disease leaf images, and achieved a significant accuracy rate [10–13]. Ashwinkumar [14] proposed an optimal mobile network-based CNN (OMNCNN) for detecting and classifying plant leaf diseases. It involves bilateral filtering-based image preprocessing and Kapur's thresholding-based image segmentation to detect the affected portions of the leaf image. U-Net is a relatively simple and widely used image semantic segmentation model and has achieved remarkable performance in medical image segmentation. However, its segmentation performance for very multi-scale small targets may be poor. U-Net can be improved from many aspects, such as encoder number, convolution operation, up-sampling and down-sampling operation, residual operation, attention mechanism, multi-scale convolution, model optimization strategy and connection type between encoding and decoding layers [15,16]. Tarasiewicz et al. [17] proposed a lightweight U-Net (LWU-Net) and applied it to multi-mode magnetic resonance brain tumor image segmentation, obtaining accurate brain tumor contour. Xiong et al. [18] proposed a multi-scale feature fusion attention U-Net (AU-Net) to improve the defect detection accuracy caused by large background noise, unpredictable environments, and different defect shapes and sizes in defect images of industrial parts. This model combines attention U-Net with a multi-scale feature fusion module to detect the defects in low-noise images effectively. Yuan et al. [19] presented an improved AU-Net, which can integrate deep and rich semantic information and shallow detail information to perform adaptive and accurate segmentation of aneurysm images with large size differences in MRI angiography. Multi-scale U-Net (MSU-Net) can concatenate the fixed and moving images with multi-scale input or image pyramid and concatenate them with corresponding layers of the same size in U-Net [20]. Tian et al. [21] proposed a modified MSU-Net with dilated convolution structure, squeeze excitation block and spatial transformer layers. Experiment results indicated that it is competitive for normal and abnormal images. Wang et al. [22] proposed an improved U-Net namely HDA-ResUNet with residual connections, adding a plug-and-play, portable channel attention block and a hybrid dilated attention convolutional layer. It makes full use of the advantages of U-Net, attention mechanism and extended convolution, and performs accurate and effective medical image segmentation for different tasks. In U-Net, some related discriminant features may be lost in image segmentation.

Inspired by LWU-Net, AU-Net and MSU-Net, a multi-scale dilated U-Net (LWMSDU-Net) is constructed to improve the performance of CDLIS. It is lightweight, and the dilated convolutional coding operation is used to fuse features from different sizes of receptive fields. The main contributions of this paper are as follows:

- LWMSDU-Net is constructed by retaining local and multi-scale detail information;
- Dilated convolution is introduced into U-Net to enlarge the receptive field of the convolution layer, improve the feature learning ability of U-Net, and obtain more information about leaf spot image;
- A residual path (Respath) connection instead of the skip connection is employed to allow gradient information to flow better through the network and overcome gradient vanishing and degradation.

The rest of this paper is arranged as follows. Section 2 introduces the related works. LWMSDU-Net is described in detail in Section 3. A lot of experiments are conducted on a crop disease leaf image dataset in Section 4. Finally, the paper is concluded and the future work is given in Section 5.

## 2. Related Works

### 2.1. Residual Block

The difference between the residual convolution block and the standard convolution block is that there is a skip connection [23]. Skip connection can effectively reduce the problems of gradient vanishing and network model degradation. Residual is the difference between the predicted value and the observed value. Suppose the first layer of the network is described as $Y = H(x)$, and a residual block of the residual network is noted as $H(x) = F(x) + x$, then $F(x) = H(x) - x$, and $y = x$ is the observed value and $H(x)$ is the predicted value, $H(x) - x$ or $F(x)$ is the residual, so it is also called the residual network.

### 2.2. Dilated Convolution

The basic principle of dilated convolution is to fill 0 in the middle of the convolution kernel to expand the receptive field as a principle, which is shown in Figure 2. By setting different expansion rates for each layer, multi-scale convolution domains can be obtained, thus obtaining multi-scale features. Its advantage is that the receptive field is enlarged without loss of features by pooling, so that each convolution output contains a wide range of features. Figure 2a corresponds to a 1-dilated convolution of 3 × 3, which is the same as an ordinary convolution operation without filling 0. Figure 2b corresponds to a 2-dilated convolution of 3 × 3. The actual convolution kernel size is still 3 × 3, but the void is 1, that is, for a 7 × 7 image patch, only 9 red points have convolution operation with a 3 × 3 kernel, and the rest points are skipped. Figure 2c is a 4-dilated convolution operation reaching the receptive field of 15 × 15. Compared with the traditional convolution operation, when the convolution of 3 layers and 3 × 3 is added together, the stride is 1, and the receptive field can only reach (kernel-1) × layer + 1 = 7, that is, the receptive field of dilated convolution increases exponentially. The corresponding convolutional images are shown in Figure 3.

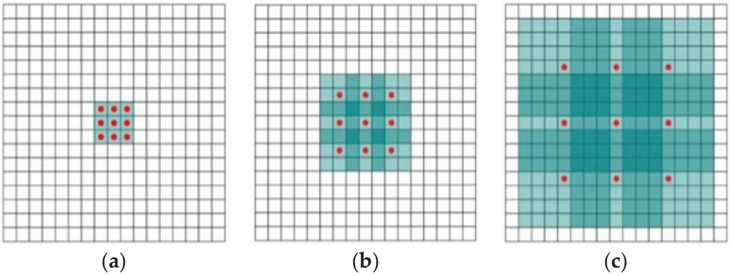


**Figure 2.** Dilated convolution kernel: (**a**) rate = 1; (**b**) rate = 2; (**c**) rate = 4.

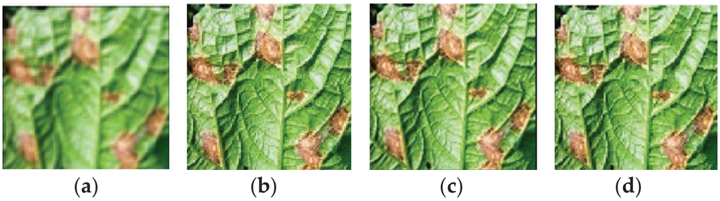


**Figure 3.** Dilated convolution images: (**a**) original image; (**b**) rate = 1; (**c**) rate = 2; (**d**) rate = 3.

### 2.3. U-Net

U-Net consists of a mutually symmetrical encoding subnetwork, decoding subnetwork and the skip connection. Its basic architecture is shown in Figure 4.

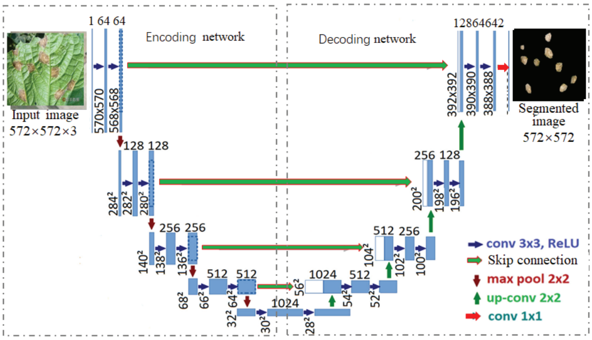


**Figure 4.** U-Net architecture.

Encoding subnetwork consists of four down-sampling operations and middle-layer operations, and each down-sampling operation includes Conv 3 × 3, BN, ReLU, MaxPool 2 × 2, DEA, where Conv 3 × 3 is 3 × 3 convolution for feature extraction, BN is a batch normalization layer to alleviate the problem of gradient disappearance, ReLU is the activation layer used to introduce nonlinear factors and accelerate network convergence, and MaxPool 2 × 2 is the maximum pooling layer of 2 × 2 to extract semantic information. Decoding subnetwork takes the output of the coding subnetwork as the input and carries out three upsampling operations, which are described as upconv 2 × 2 + Copy&crop + Conv 3 × 3 + BN + ReLU + DE module, where upconv 2 × 2 is a 2 × 2 upsampling convolution operation used to restore the size and size of the feature maps, and copy&crop, namely skip connection, refers to integrate the rough features of the encoding subnetwork with the refined features of the decoding to better retain the spatial information and detail information of the original image and then improve the image accuracy.

### *2.4. Summarization*

The characteristics of the Residual block, dilated convolution and U-Net are summarized as follows.

Residual blocks can increase the depth of the network, help solve the problems of gradient disappearance and gradient explosion, and ensure good performance while training deeper networks.

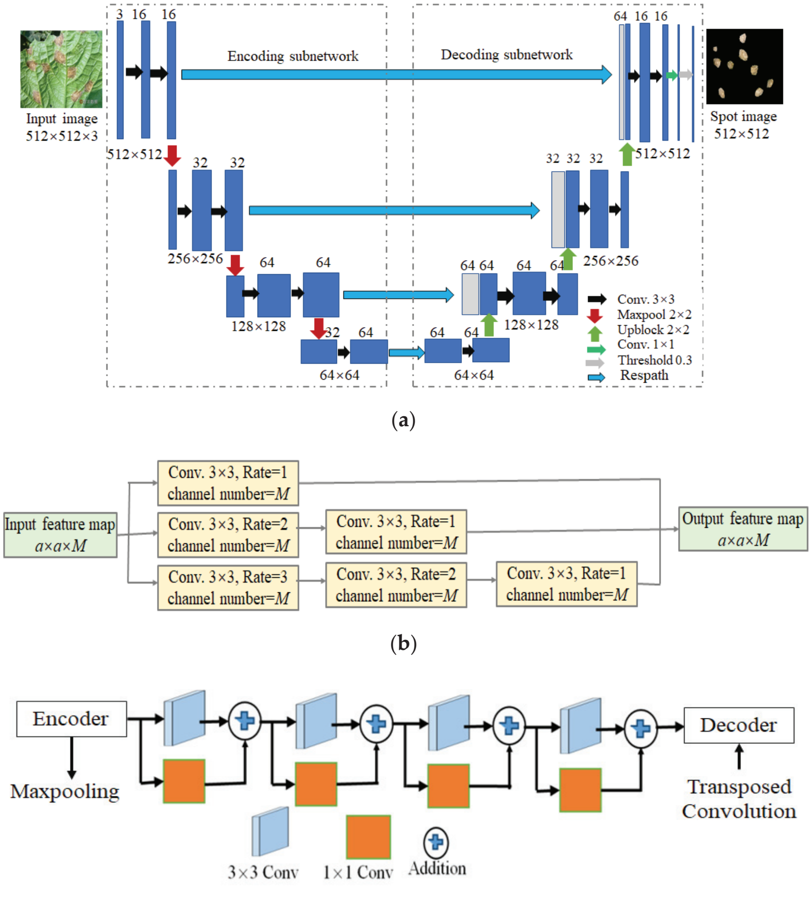
When the network layer requires a large receptive field, but the computing resources are limited and cannot increase the number or size of convolution kernels, dilated convolution can be considered. Its advantages are that the receptive field can be increased without pooling information, so that each convolution output contains a large range of information. However, the dilated convolution may have a grid effect, that is, the convolution kernels are discontinuous; if only multiple 3 × 3 convolution kernels with dilation rate = 2 are stacked multiple times, not all input pixels are calculated. The key to designing a good dilated convolution layer is how to deal with the relationship between objects of different sizes at the same time.

U-net can provide context semantic information of segmentation target in the whole image, train end-to-end from a few images, and is superior to the previous sliding window convolution network. It uses features spliced together in the channel dimension to form thicker features, which can provide finer features for image segmentation. The addition of corresponding points used in FCN fusion does not form thicker features.

## 3. Lightweight Multi-Scale Dilated U-Net (LWMSDU-Net)

### *3.1. LWMSDU-Net Architecture*

Although many improved U-Net models have been constructed and achieved remarkable results, they do not take into account the number of trainable parameters, the calculation of the model, and the characteristics of the disease leaf image shown in Figure 1, and are not suitable for deployment on devices with limited computing power and storage space. To improve the accuracy and effectiveness of CDLIS, a lightweight multi-scale dilated U-Net (LWMSDU-Net) is constructed for CDLIS by making use of the advantages of lightweight, multi-scale, residual convolution, and dilated U-Net. Its architecture is shown in Figure 5a.



(**a**)

(**b**)

(**c**)

**Figure 5.** The architecture of LWMSDU-Net: (**a**) LWMSDU-Net structure; (**b**) dilated structure, where the size of the feature map is $a \times a$, and M is the channel number; (**c**) Respath structure.

The multi-scale dilated convolution is employed instead of the convolution of U-Net in the convolutional layer. According to the characteristics of the diseased leaf image, the expansion rates r is set as 1, 2 and 3, respectively, so that the irregular disease regions of different scales could be effectively segmented and the overall segmentation performance could be improved. The structure of multi-scale dilated convolution is shown in Figure 5b. Suppose $f_h$ represents the height of the original convolution kernel and $f_w$ is the width of the original convolution kernel, the height of the effective convolution kernel thus obtained is $f_h + (f_h - 1)(r - 1)$, and the width is $f_w + (f_w - 1)(r - 1)$.

In U-Net, the skip connection is used to connect the encoder and decoder. It is simple to implement, but there is often a big semantic gap in semantic between the encoder and decoder due to the complex disease leaf images. To improve segmentation results and relieve this semantic gap, a residual path (Respath) instead of the skip connection is constructed to connect the encoder and decoder before concatenating, so that the encoder features perform some additional convolution operations before being spliced with the corresponding features in the decoder. Respath structure is shown in Figure 5c, consisting of four residual convolution blocks [22].

### *3.2. Process of CDLIS*

The steps of LWMSDU-Net based CDLIS method include training stage and test stage. The original parameters of LWMSDU-Net are set by transfer learning, then the training dataset is used to optimize its parameters iteratively, and the test set is used to verify the model recognition effect. Model training is the most crucial step in the experiment because the trained appropriate model can improve the classification accuracy, and the experiment mode and hyper-parameter configuration of this paper are standardized to ensure the validity of the experiment. In the model training stage, to enhance the model image feature extraction ability and training speed, the PlantVillage dataset is used as the input of LWMSDU-NET, and the parameters of pre-training are retained. Then, the network model after pre-training is trained by the constructed augmented dataset of maize corn cucumber diseases. Pre-training can accelerate the model training speed, effectively enhance the fitting ability of the network, and improve the accuracy of CDLIS on the limited dataset.

The training stage includes the following steps:

Step 1: Convert the disease leaf images from R*G*B color space to L*a*b, and using the simple linear iterative cluster (SLIC) method to preprocess the transformed disease leaf images;

Step 2: Disease leaf images are converted to TensorFlow2 format, divided into different batches and then input into LWMSDU-Net for feature extraction (https://github.com/tzutalin/labeling/releases, accessed on 7 October 2022);

Step 3: Use transfer learning to reduce the number of training iterations and speed up training the network;

Step 4: Fuse the extracted features from LWMSDU-Net, and input the fused features into the classifier for training the classifier;

Step 5: If the error between the authentic labeled training images and the predictive labeled training images is more than the given threshold, go back to Step 2 and further train LWMSDU-Net. Otherwise, the training stage is stopped.

The test stage includes the following steps:

Step 1: Normalize the scale of the test images;

Step 2: Put the normalized images into the trained LWMSDU-Net and extract features;

Step 3: Fuse the extracted features and then put them into the SoftMax classifier;

Step 4: Output the recognition result of the input image.

## 4. Experiments and Analysis

In this section, a lot of experiments of CDLIS are conducted to validate the proposed method. Comparative experiments and results are then analyzed and discussed. All experiments are carried out: Windows 7 64-bit operating system, Intel Xeon E5-2643v3 @3.40 GHz CPU, 64 GB RAM, NVidia Quadro M4000 GPU, 8 GB of video memory, by CUDA Toolkit 9.0, CUDNN V7.0, Python 3.5.2, Tensorflow-GPU 1.8.0 with Keras open source deep learning framework. In LWMSDU-Net, the initial weight parameters are set randomly, the number of iterations is set as 500, the initial learning rate is specified as 0.001 and then gradually reduced to 0.1 times in training stages, the momentum is set as 0.99 to reduce the overfitting problem, the weight decay is set as 0.005, and the training images are divided into 10 batches and sent to the network model in turn. To improve the segmentation effect of the model, LWMSDU-Net is trained 1200 rounds with each round of iteration of 3000 times, and the widely used stochastic gradient descent (SGD) is used as a training mechanism. Since the last layer of the network is the Softmax classifier, Softmax-loss is used as a loss function, which is more stable in computing. Other parameters are set as the default parameters of the U-Net framework. The trained model is evaluated by the verification images. In LWMSDU-Net, all RGB images of disease leaf are preprocessed through median filtering and then standardized by cropping to reduce calculation and training time. Each image is normalized and cropped to a size of 512 × 512 pixels.

### *4.1. Dataset*

PlantVillage (https://tensorflow.google.cn/datasets/catalog/plant_village, accessed on 7 October 2022) is an open source dataset. It was collected at experimental research stations associated with Land Grant Universities in the USA (Penn State, Florida State, Cornell and others). It is an open source dataset for diagnosing and recognizing crop diseases. It consists of 54,303 healthy and unhealthy leaf images of 26 diseases of 14 crops taken in the natural environment of farmland. In this paper, it is utilized for pre-training to make up for the shortage of the training samples. The pre-trained model is then trained and tested using the real crop dataset.

In this paper, five types of maize and cucumber disease were taken with digital cameras, smart phones and other devices in the Yangling Agricultural Demonstration Field, Shanxi Province, including two corn leaf images of blight and brown spot, and three cucumber leaf images of target spot, brown spot and anthracnose, 20 leaf images for each disease. As the disease leaf images vary with crop growth environment, background, sunshine and photographic equipment, to reflect the real scene and improve the generalization ability of the model, all images were taken in the morning, noon, afternoon, sunny and cloudy days from April to June 2021. Five disease leaf image samples are shown in Figure 6.

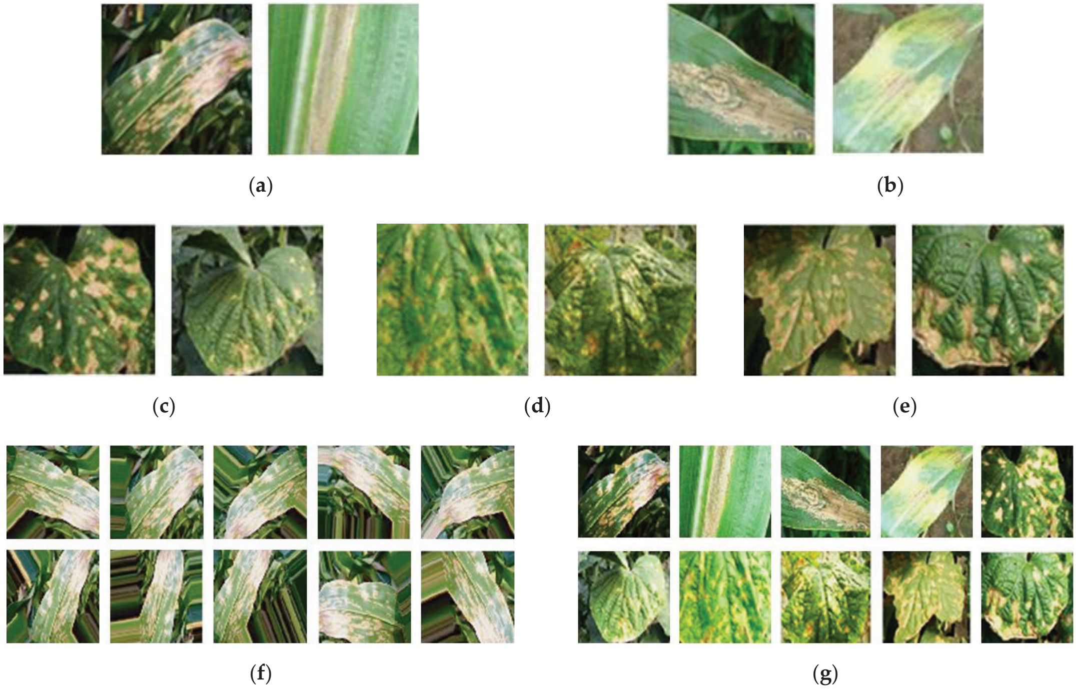


**Figure 6.** Five typical disease leaf images: (**a**) Original images for leaf blight of maize; (**b**) original images for brown blotch of maize; (**c**) original images for target spot of cucumber; (**d**) original images for brown spot of cucumber; (**e**) original images for anthrax disease of cucumber; (**f**) 10 augmented images of a maize disease leaf image; (**g**) equalized images of the above images in the above (**a**–**e**).

The number of the collected disease leaf images is limited, which easily leads to the overfitting. Augmenting algorithms, such as randomly enhanced lighting, randomly cropping, rotation, shifting, adding random noise and mirroring, are often used to augment the number of training samples. Augmenting operation can enlarge the diversity of the training samples and avoid overfitting. In the following experiments, each image is augmented to 10 images, as shown in Figure 6f. An augmented dataset containing 1100 images is constructed, including 100 original and 1000 augmented images. The details of the original dataset and its augmented dataset are shown in Table 1.

**Table 1.** The details of the original dataset and its augmented dataset.

| Disease Type | | Number of Original Images | Number of Augmented Images | Total |
|---|---|---|---|---|
| Corn | Leaf blight | 20 | 200 | 220 |
| | Brown spot | 20 | 200 | 220 |
| Cucumber | Target spot | 20 | 200 | 220 |
| | Brown spot | 20 | 200 | 220 |
| | Anthracnose | 20 | 200 | 220 |
| Total number of images | | 100 | 1000 | 1100 |

To reduce environmental noise and computational complexity, smooth the image, remove salt and pepper noise and retain image edge information, the median filtering algorithm is carried out on the crop disease leaf image, as follows:

$$y(n) = med[x(i-N), \ldots, x(i+N)] \tag{1}$$

where $x(i)$ is the value of the pixel point in the center of the sliding window, *med* is the value of the pixel's neighborhood, and $y(n)$ is the median filtering output value.

From Figure 6g, it is observed that median filtering can enhance the contrast of the disease leaf images and the filtered images can significantly characterize the disease leaf image features. The image recognition accuracy of CDLIS can be improved after median filtering.

The effective disease leaf image blocks are cropped from the collected images to reduce the influence of complex background on CDLIS, and the leaf images are uniformly processed into 512 × 512 resolution images. Secondly, Labelme is used to label the image set of crop disease leaves in the demonstration base. Each image contains two data labels: 1 represents the area of crop leaf disease spots, and 0 represents the background. Annotation data are stored in JSON format, and the command of labelme json to the dataset is used to convert data labels into binarized PNG graphs. The color annotated image can be obtained by multiplying the original and binarized images. The cropping and annotating process is shown in Figure 7.

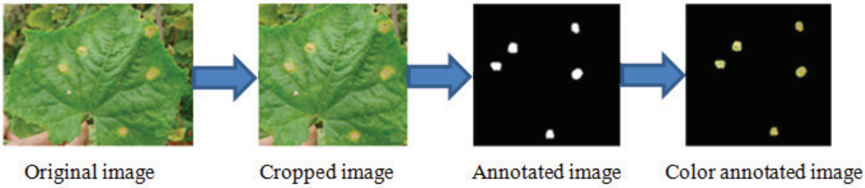


**Figure 7.** The cropping and annotating process.

In order to reduce the influence of geometric transformation and accelerate the speed of gradient descent to find the optimal solution, each image is normalized, which is implemented by mapping the pixel value of the image to (0,1) by linear function transformation,

$$y = (x\text{-MinValue})/(\text{MaxValue-MinValue}) \tag{2}$$

where $x$ and $y$ are the values before and after conversion, respectively, and MaxValue and MinValue are the maximum and minimum values of the sample, respectively.

There are some methods to form the statistical tests [24]. In the paper, a five-fold cross verification scheme is employed to validate LWMSDU-Net, that is, all 1100 leaf images per disease are randomly divided into five subsets with the same number of images, each is used as a test set for testing the model, and the remaining images are used as training samples for training the model. Each subset is taken as a test set once, and a total of five tests are conducted. The average segmentation result of five times experiments is the final result.

### 4.2. Results

Average precision, average recall and average $F_1$-score of five-fold cross verification experiments are often adopted to test network performance, and are calculated as follows:

$$Recall = \frac{B_{seg}}{B_{seg} + I_{unseg}} \tag{3}$$

$$Precision = \frac{B_{seg}}{B_{seg} + I_{wseg}} \tag{4}$$

$$F_1\text{-score} = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$

where $B_{seg}$ is the pixel number correctly segmented into spot pixels, $I_{unseg}$ is the pixel number not segmented into spot pixels but being spot pixels in the image, and $I_{wseg}$ is the pixel number that segments the background pixels into spot pixels.

Pixel accuracy $Acc_{Pixel}$ is often used to evaluate the performance of the model. It is the total number of pixels whose real pixel category is predicted as a category, which is calculated as follows,

$$Acc_{Pixel} = \frac{1}{m}\sum_{i=1}^{m} f_i, f_i = \left\{ \begin{array}{l} 1, \left|y_i - y_i'\right| < T \\ 1, \left|y_i - y_i'\right| \geq T \end{array} \right. \tag{6}$$

where $y_i$ is the $i$th real pixel category, and $y_i'$ is the $i$th predicted category, $T$ is a threshold.

In fact, the final output of the image segmentation models is a grayscale image and the values of all pixels vary from 0 to 1, $T$ is often set 0.5.

LWMSDU-Net is trained on the augmented dataset. The training accuracy and loss are recorded after each iteration, as shown in Figure 8. It can be seen from Figure 8 that with the increasing number of training iterations, the accuracy of the model keeps rising while the loss value keeps decreasing. When the number of iterations reaches 2500, the accuracy is stable at 0.91, the fluctuation is stable within 1 percentage point, the loss value is stable at 0.043, and the fluctuation is within 0.01. The model has high accuracy and good robustness. It can be observed from the analysis that the LWMSDU-Net in this paper is effective and feasible for CDLIS.

The pre-trained model on the PlantVillage dataset is trained in the constructed dataset. In order to test the training performance of LWMSDU-Net, it is compared with U-Net, LWU-Net [17], AU-Net [18] and MSU-Net [21] on the augmented dataset. Each of the three improved models has its advantages, where LWU-Net is a lightweight U-Net, AU-Net takes advantage of attention, and MSU-Net is a multi-scale U-Net. Figure 9 shows their segmentation accuracies versus the number of iterations in the convergence process, where all models are pre-trained on PlantVillage dataset. From Figure 9, it is observed that all loss values of five network models drop rapidly before the 1000th iteration, and are nearly stable after the 1500th iteration. From Figure 9, it is also found that LWMSDU-Net outperforms other four models and achieves the best convergence performance after the 2700th iteration. The reason may be that dilated convolution and Respath are used to speed up its training and improve its segmenting performance. Comparing Figures 9 and 10, it can be found that the performance of LWMSDU-Net after pre-training is very good.

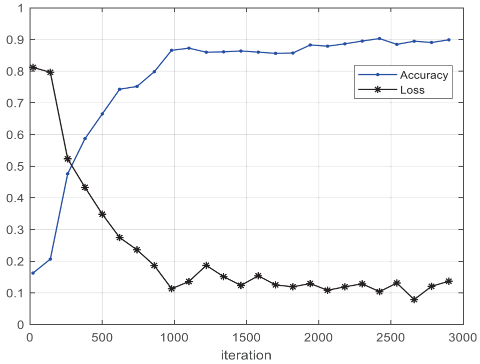


**Figure 8.** Accuracy and loss value versus iteration.

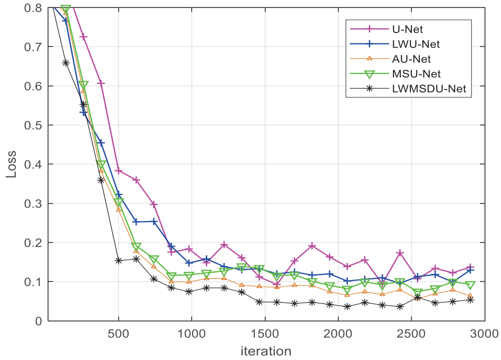


**Figure 9.** Segmentation accuracy versus the number of iterations of four networks.

To be fair, 5 trained models are chosen after the 3000th iteration. The typical segmented disease leaf images of five models are shown in Figure 10.

From Figures 9 and 10, it is observed that all four modified U-Net models are much better than the traditional U-Net. In five-fold cross verification experiments, the trained U-Net, LWU-Net, AU-Net, MSU-Net and LWMSDU-Net are used to segment the disease leaf images of the augmented dataset, and their segmentation results are shown in Table 2.

**Table 2.** Segmentation results of U-Net, LWU-Net, AU-Net, MSU-Net and LWMSDU-Net.

| Method | U-Net | LWU-Net | AU-Net | MSU-Net | LWMSDU-Net |
|---|---|---|---|---|---|
| Precision | 86.13 | 89.86 | 92.54 | 93.25 | 94.18 |
| Recall | 82.36 | 81.18 | 84.31 | 85.25 | 89.10 |
| $F_1$-score | 84.20 | 85.30 | 88.23 | 89.07 | 91.57 |
| Pixel accuracy | 85.66 | 90.24 | 91.50 | 91.45 | 93.71 |
| Training Time | 12.51 h | 6.42 h | 10.52 h | 11.14 h | 5.17 h |
| Testing time | 5.64 s | 5.18 s | 5.42 s | 4.85 s | 4.73 s |

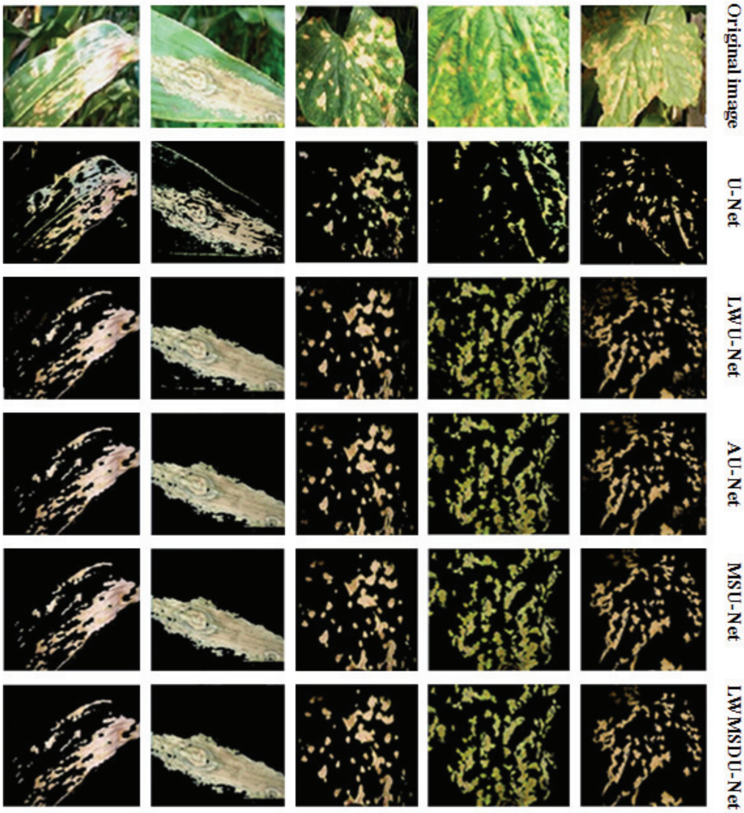


**Figure 10.** Typical segmented disease leaf images by 5 models.

### 4.3. Ablation Experiments and Results

The proposed model LWMSDU-Net is based on U-Net, and makes use of the characteristics of the Respath connection, dilated convolution and multi-scale Inception module. To verify the effectiveness of their combination, some ablation experiments are carried out. The experimental results are shown in Table 3 by combining different convolution structures and connection structures, where U-Net employs 3 × 3 convolution and skip connection, Res-U-Net is combined by U-Net and residual block for image segmentation [25], and Inception U-Net consists of a normalization layer, convolution layers, and Inception layers (concatenated 1 × 1, 3 × 3, and 5 × 5 convolution [26].

**Table 3.** Segmentation results by different combinations of convolution and connection.

| Combination Mode | Precision | Training Time |
|---|---|---|
| U-Net: 3 × 3 conv.+ Skip connection | 86.13 | 12.51 h |
| U-Net: 3 × 3 conv. + Respath connection | 87.22 | 11.36 h |
| Res-U-Net: residual block + Skip connection | 90.14 | 11.75 h |
| Inception U-Net: Inception + Skip connection | 92.16 | 10.46 h |
| U-Net: Inception module + Respath connection | 91.57 | 9.73 h |
| U-Net: dilated Inception module + skip connection | 92.46 | 7.13 h |
| LWMSDU-Net: dilated Inception + Respath connection | 94.18 | 5.17 h |

From Table 3, it is found that the proposed LWMSDU-Net exhibits quite significant results as compared to the original U-Net, Inception U-Net and different combination architecture, and the results validate the effectiveness of dilated Inception module, Respath connection and their combination.

## 5. Analysis and Discussion

From Figures 9 and 10 and Tables 2 and 3, it is observed that LWMSDU-Net and other modified U-Net networks can obtain more detailed spot images even if the spots are small and not clearly contrasted with the healthy leaf areas and background, and specially, LWMSDU-Net is superior to the other models in accuracy and computing complexity. LWU-Net and LWMSDU-Net have shorter training times because they are lightweight and have fewer trainable parameters, while LWMSDU-Net has the shortest training time because it utilizes dilated convolution and Respath connection. U-Net splices the features together in the channel dimension to form richer segmentation features. U-Net can completely segment the lesion area including the small lesion area, but it cannot effectively divide the adhesion lesion, resulting in more missing lesion pixels. CDLIS by U-Net has some false positive areas, which could not distinguish the lesion area from the background. CDLIS by its modified models is better than that of U-Net. MSUN-Net is slightly better than LWU-Net and AU-Net due to the multi-scale convolution. AU-Net is slightly superior to LWU-Net because of the attention mechanism. LWMSDU-Net can accurately segment the disease leaf images including the lesion area and the edge details of the lesion, due to it utilizing Respath instead of the skip connection of U-Net, and dilated convolution instead of convolution. It is indicated that Respath and dilated convolution can improve the performance of CDLIS.

Compared with other networks, the experimental results demonstrate that LWMSDU-Net achieves a significant segmentation effect. However, it is validated only on a single enhanced dataset. The super-parameters of the training network need to be adjusted according to the dataset being processed, so it cannot completely guarantee that the model weight parameters can be transmitted to other data sets.

In terms of the memory occupied by the model, VGG16 occupies the largest memory, 552.0 MB, the memory occupied by AlexNet is 227.6 MB, because the number of parameters of the fully connected layers is the largest in the entire model. In deep CNN, in order to increase receptive fields and reduce the amount of computation, it is always necessary to conduct downsampling (pooling or s2/conv). In this way, although the receptive fields can be increased, the spatial resolution is reduced. In order not to lose resolution and still expand the receptive field, dilated convolution can be used. By adding zeros to expand the receptive field, the original 3 × 3 convolution kernel can have a 5 × 5 (dilated rate = 2) or a larger receptive field under the same parameter amount and calculation amount, so that no down sampling is required. Dilated convolution introduces only a parameter called dilated rate to the convolution layer, which defines the distance between values when the convolution kernel processes data. In other words, compared with the original standard convolution, the extended convolution has an additional division rate parameter. The division rate of a normal revolution is 1. It can be observed that the number of parameters of the dilated convolution is greatly reduced. Based on this, dilated convolution is added to U-Net, which effectively reduces the number of model parameters. The number of parameters of U-Net is 7.76 M, while the number of parameters of this model after training is 5.8 MB.

## 6. Conclusions

Aiming at the problem of crop disease leaf image segmentation (CDLIS), the traditional U-Net model is improved by making use of dilated convolution and Respath. Multi-scale dilated convolution instead of traditional convolution is used to increase the receptive field and improve the feature learning ability of U-Net. Respath instead of skip connection between Encoder and Decode is utilized to concatenate the lesion information of disease leaf image. PlantVillage is employed for pre-training to make up for the shortage of the training samples, overcome the overfitting problem and improve the network performance.

The proposed CDLIS method based on LWMDU-Net can be applied to the actual agricultural environment, help farmers quickly and accurately detect crop diseases, and provide effective technical means for scientific disease control. For future work, it is necessary to further verify and optimize the model and construct a more lightweight version for deployment to personal computers and smartphones.

**Author Contributions:** Conceptualization, C.X. and S.Z.; methodology, C.X.; software, C.X. and C.Y.; formal analysis, C.X. and C.Y.; writing—original draft preparation, C.X.; writing—review and editing, S.Z. 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 (Nos. 62172338 and 62072378).

**Data Availability Statement:** Not applicable.

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

## References

1. Sharma, V.; Tripathi, A.K.; Mittal, H. Technological Advancements in Automated Crop Pest and Disease Detection: A Review & Ongoing Research. International Conference on Computing, Communication. In Proceedings of the 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS), Kochi, India, 23–25 June 2022. [CrossRef]
2. Hussain, N.; Khan, M.A.; Tariq, U.; Kadry, S.; Yar, M.A.E.; Mostafa, A.M.; Alnuaim, A.A.; Ahmad, S. Multiclass Cucumber Leaf Diseases Recognition Using Best Feature Selection. *Comput. Mater. Contin.* **2022**, *2*, 3281–3294. [CrossRef]
3. Praveen, P.; Nischitha, M.; Supriya, C.; Yogitha, M.; Suryanandh, A. To Detect Plant Disease Identification on Leaf Using Machine Learning Algorithms. In *Intelligent System Design*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 239–249. [CrossRef]
4. Huo, M.; Tan, J. Overview: Research Progress on Pest and Disease Identification. In *Pattern Recognition and Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 404–415.
5. Wang, Z.; Wang, K.; Pan, S.; Han, Y. Segmentation of Crop Disease Images with an Improved K-means Clustering Algorithm. *Appl. Eng. Agric.* **2018**, *34*, 277–289. [CrossRef]
6. Fan, X.; Luo, P.; Mu, Y.; Zhou, R.; Tjahjadi, T.; Ren, Y. Leaf image based plant disease identification using transfer learning and feature fusion. *Comput. Electron. Agric.* **2022**, *196*, 106892. [CrossRef]
7. Singh, A.K.; Sreenivasu, S.; Mahalaxmi, U.; Sharma, H.; Patil, D.; Asenso, E. Hybrid Feature-Based Disease Detection in Plant Leaf Using Convolutional Neural Network, Bayesian Optimized SVM, and Random Forest Classifier. *Hindawi J. Food Qual.* **2022**, *2020*, 2845320. [CrossRef]
8. Liu, X.; Bai, X.; Wang, L.; Ren, B.; Lu, S. Review and Trend Analysis of Knowledge Graphs for Crop Pest and Diseases. *IEEE Access* **2019**, *7*, 62251–62264.
9. Fernandez, C.I.; Leblon, B.; Wang, J.; Haddadi, A.; Wang, K. Cucumber powdery mildew detection using hyperspectral data. *Can. J. Plant Sci.* **2022**, *1*, 20–32. [CrossRef]
10. Ahmad, J.; Jan, B.; Farman, H.; Ahmad, W.; Ullah, A. Disease Detection in Plum Using Convolutional Neural Network under True Field Conditions. *Sensors* **2020**, *20*, 5569. [CrossRef] [PubMed]
11. Zhang, J.; Yang, Y.; Shao, K.; Bai, X.; Fang, M.; Shan, G.; Chen, M. Fully convolutional network-based multi-output model for automatic segmentation of organs at risk in thorax. *Sci. Prog.* **2021**, *104*, 1–19. [CrossRef] [PubMed]
12. Bhattacharya, S.; Mukherjee, A.; Phadikar, S. A Deep Learning Approach for the Classification of Rice Leaf Diseases. In *Intelligence Enabled Research*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 61–69.
13. Zeng, W.; Li, H.; Hu, G.; Liang, D. Lightweight dense-scale network (LDSNet) for corn leaf disease identification. *Comput. Electron. Agric.* **2022**, *197*, 106943. [CrossRef]
14. Ashwinkumar, S.; Rajagopal, S.; Manimaran, V.; Jegajoth, B. Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. *Mater. Today Proc.* **2022**, *51*, 480–487. [CrossRef]
15. Han, Y.; Ye, J.C. Framing U-Net via Deep Convolutional Framelets: Application to Sparse-View CT. *IEEE Trans. Med. Imaging* **2018**, *37*, 1418–1429. [CrossRef] [PubMed]
16. Li, Q.; Jia, W.; Sun, M.; Hou, S.; Zheng, Y. A novel green apple segmentation algorithm based on ensemble U-Net under complex orchard environment. *Comput. Electron. Agric.* **2021**, *180*, 105900. [CrossRef]
17. Tarasiewicz, T.; Kawulok, M.; Nalepa, J. Lightweight U-Nets for Brain Tumor Segmentation. In *International MICCAI Brainlesion Workshop*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12659, pp. 3–14.
18. Xiong, Y.J.; Gao, Y.B.; Wu, H.; Yao, Y. Attention U-Net with Feature Fusion Module for Robust Defect Detection. *J. Circuits Syst. Comput.* **2021**, *31*, 2150272. [CrossRef]
19. Yuan, W.; Peng, Y.; Guo, Y.; Ren, Y.; Xue, Q. DCAU-Net: Dense convolutional attention U-Net for segmentation of intracranial aneurysm images. *Vis. Comput. Ind. Biomed. Art* **2022**, *5*, 9. [CrossRef] [PubMed]
20. Li, S.; Zheng, J.; Li, D. Precise segmentation of non-enhanced computed tomography in patients with ischemic stroke based on multi-scale U-Net deep network model. *Comput. Methods Programs Biomed.* **2021**, *208*, 106278. [CrossRef] [PubMed]

21. Tian, Y.; Hu, Y.; Ma, Y.; Ma, H.; Liu, J. Multi-scale U-net with Edge Guidance for Multimodal Retinal Image Deformable Registration. In Proceedings of the 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 1360–1363. [CrossRef]
22. Wang, Z.; Zou, Y.; Liu, P.X. Hybrid Dilation and Attention Residual U-Net for Medical Image Segmentation. *Comput. Biol. Med.* **2021**, *134*, 104449. [CrossRef] [PubMed]
23. Fu, L.; Li, S.; Sun, Y. Lightweight Convolutional Neural Network for Apple Leaf Disease Identification. *Front. Plant Sci.* **2022**, *13*, 831219. [CrossRef] [PubMed]
24. Ibrahim, A.; Seyedali, M.; Mohammed, E.; Sherif, S.M.G.; Mosleh, M.A.; Tare, F.I.; EI-Sayed, M.E. Wind speed ensemble forecasting based on deep learning using adaptive dynamic optimization algorithm. *IEEE Access.* **2021**, *9*, 125787–125804. [CrossRef]
25. Mustafa, N.; Zhao, J.; Liu, Z.; Zhang, Z.; Yu, W. Iron ORE Region Segmentation Using High-Resolution Remote Sensing Images Based on Res-U-Net. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Piscataway Township, NJ, USA, 2020; pp. 2563–2566. [CrossRef]
26. Punn, N.S.; Agarwal, S. Inception U-Net Architecture for Semantic Segmentation to Identify Nuclei in Microscopy Cell Images. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2020**, *16*, 1–15. [CrossRef]

*electronics*

*Article*

# Human Perception Intelligent Analysis Based on EEG Signals

**Bingrui Geng [1,*], Ke Liu [2] and Yiping Duan [3]**

[1] School of Information and Communication Engineering, Communication University of China, Beijing 100024, China
[2] Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China
[3] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
* Correspondence: gengbr@cuc.edu.cn

**Abstract:** The research on brain cognition provides theoretical support for intelligence and cognition in computational intelligence, and it is further applied in various fields of scientific and technological innovation, production and life. Use of the 5G network and intelligent terminals has also brought diversified experiences to users. This paper studies human perception and cognition in the quality of experience (QoE) through audio noise. It proposes a novel method to study the relationship between human perception and audio noise intensity using electroencephalogram (EEG) signals. This kind of physiological signal can be used to analyze the user's cognitive process through transformation and feature calculation, so as to overcome the deficiency of traditional subjective evaluation. Experimental and analytical results show that the EEG signals in frequency domain can be used for feature learning and calculation to measure changes in user-perceived audio noise intensity. In the experiment, the user's noise tolerance limit for different audio scenarios varies greatly. The noise power spectral density of soothing audio is 0.001–0.005, and the noise spectral density of urgent audio is 0.03. The intensity of information flow in the corresponding brain regions increases by more than 10%. The proposed method explores the possibility of using EEG signals and computational intelligence to measure audio perception quality. In addition, the analysis of the intensity of information flow in different brain regions invoked by different tasks can also be used to study the theoretical basis of computational intelligence.

**Keywords:** computational intelligence; quality of experience; human perception; electroencephalogram

**Citation:** Geng, B.; Liu, K.; Duan, Y. Human Perception Intelligent Analysis Based on EEG Signals. *Electronics* **2022**, *11*, 3774. https://doi.org/10.3390/electronics11223774

Academic Editor: Akshya Swain

Received: 7 October 2022
Accepted: 15 November 2022
Published: 17 November 2022

## 1. Introduction

With the continuous development of computer technology, how to deal with and analyze the potentially insightful information in big data has become an extremely urgent problem that must be overcome. The emergence of computational intelligence and artificial intelligence technology has become an effective way to solve the above problems in various scientific fields. Many outstanding works have further promoted the application of computational intelligence. In the field of image analysis, machine learning (ML) and deep neural networks are used for feature extraction and image segmentation [1,2].

In the field of multimedia communication, with the development of multimedia and communication technology, new services and applications emerge in an endless stream. There are more and more ways for people to obtain information through various terminals, and the audio–visual forms are becoming increasingly abundant; traditional audio, video and emerging virtual reality, augmented reality and other forms are becoming more and more convenient. Ubiquitous multimedia and converged media services are changing people's lives, which also leads to great changes in business content and data volume. Whether a product can provide users with satisfactory services has become a decisive factor for success in the rapidly changing market environment, which is crucial for communication service providers and business service providers. Under the new market demand, the communication changes from data communication to multimedia communication. User satisfaction is also affected by a variety of factors, and the mechanism of action is much more complex [3,4]. At this time, ML is often used for resource allocation, quality management and quality prediction [5].

Traditionally, the most recognized method is a technology parameter-centric quality metric named quality of service (QoS) [6], which mainly considers objective technical parameters such as jitter, packet loss, delay, etc. It has been widely used in technology and industry. Additional research has found that the key performance QoS of traditional networks measures the objective quality [6]. The QoS does not consider the actual experience of users. Therefore, a good QoS may not satisfy users, which leads to the bottleneck of improving user satisfaction [7].

International standardization organizations ITU-T [8] defined QoE as "the overall acceptability of an application or service, as perceived subjectively by the end-user" [9]. According to such a definition, the factors influencing QoE are more diverse, including not only audio quality, video quality and network quality, but also service content, multimedia devices and users' personal feelings [3]. For service providers and network operators, the shift from the traditional quality evaluation method focusing on QoS service performance to the QoE evaluation aiming at users' perception and demand seems to better reflect the original intention of providing users with better-quality services. Therefore, QoE research has become an interdisciplinary field involving a lot of knowledge, such as social psychology, cognitive science, intelligent computing and engineering science [10].

At present, the evaluation methods of QoE are mainly divided into two categories: objective parameter-based evaluation and intelligent cognitive-based subjective evaluation [4,7,11], as shown in Figure 1. The objective parameter-based evaluation method first measures or calculates the objective parameters, or establishes a mathematical estimation model from objective parameters to subjective experience, which is based on the statistical knowledge derived from a large number of data, then the estimation model is further used to transform the objective parameters into the estimated value of experience quality [11]. Both the advantages and disadvantages of this kind of method are very prominent. One advantage is that if a suitable mathematical model has been embedded in the QoE evaluation system, the evaluation of QoE will be efficient. Therefore, it is still the best choice for the actual multimedia business scenario [12]. The disadvantage is that it is impossible to truly experience the multi-level satisfaction of users without their participation. Intelligent cognitive-based subjective evaluation refers to evaluation that requires users' participation. Either the specific indicators or the information of experience quality needs to be obtained directly from users. It can be reported by users straight away or be measured by users' relevant physiological variables. These physiological data need to further adopt feature extraction and learning to calculate and analyze the real feelings of the user [7,10,13]. Based on the correlation between perceptual processes and neurophysiology, using advanced calculation and analysis of user neurophysiological indicators to quantify users' subjective experience is an important way to overcome the bias caused by users' upper cognitive behavior in the process of subjective feedback [14]. In addition, due to the amount of data and analytical requirements, computational intelligence techniques also provide more feasible methods for subjective QoE prediction and quality analysis [15,16].

In multimedia communication, the sound is the sensory channel with the highest priority, which is the basis of audiovisual perception. Nonetheless, to our knowledge, the influence of auditory perception on QoE is much less studied than that of visual perception on QoE. This paper proposed a new method to explore the possibility of measuring the user's auditory subjective feelings by collecting the physiological sensory signals from the user's central nervous system. The main contributions of our work are summarized as follows. First, a complete experiment was designed to collect perceptual data of users under different audio quality conditions, including EEG data, subjective judgment data and perceptual semantic data. Second, a new method of studying the relationship between human perception ability and audio noise intensity using EEG signals was proposed, and the perceptual tolerance of audio noise in different semantic scenarios was obtained. In addition, the relationship between audio signal to noise ratio (SNR), audio scenarios, user emotions, and noise perceptual tolerance was explored. Finally, the location of the brain area for audio processing was explored, and the connectivity of related brain regions was quantitatively analyzed.

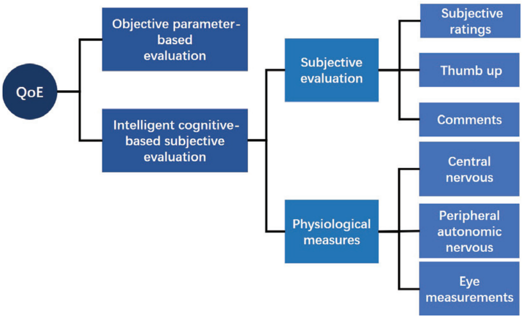

QoE
Objective parameter-based evaluation
Intelligent cognitive-based subjective evaluation
Subjective evaluation
Physiological measures
Subjective ratings
Thumb up
Comments
Central nervous
Peripheral autonomic nervous
Eye measurements

**Figure 1.** The evaluation methods of QoE.

The rest of this paper is organized as follows. Section 2 reviews related work for QoE evaluation. Section 3 briefly describes the experiment design and data recording. In Section 4, we describe the signal processing and analysis methods in detail, and Section 5 expands on the experimental results and discussion. In Section 6, we conclude the current work and give the direction for future work.

## 2. Related Work

Since the concept of QoE was proposed, there has been a lot of excellent work published continuously on QoE prediction and evaluation. In the paper [17], the authors used subjective mean opinion score (MOS) data and evolutionary algorithms to optimize QoE on a global scale. In the paper [18], deep learning (DL) was used to extract generalized features and representation learning from text data, video and audio data and classification parameters, and finally achieved QoE prediction through the classifier. The data in the above works came from communication networks and multimedia devices. Psychological and physiological data were retrieved directly from the user. The psychology aspect mainly involves the user questionnaire, the ratings, and so on. The physiology aspect mainly involves the collection and processing of users' physiological signals. Currently, physiological measures used to assess the quality of multimedia experience fall into three categories: central nervous system measurement, peripheral autonomic nervous system measurement, and eye measurements [19]. Human primary perception and thinking activities belong to the central nervous system function. The neural connections between attention, decision making, and memory in animals and humans have been described in a wide range of experimental studies [20]. Because the physiological indicators measured by the central nervous system can directly reflect human perception and other thinking activities, this method is more conducive to the calculation and analysis of users' perception and cognitive process of multimedia stimulation [14]. The most common devices available are electroencephalography (EEG) [19], near-infrared spectroscopy (NIRS) [21], functional magnetic resonance imaging (fMRI) [22] and magnetoencephalography (MEG) [23]. The activity of the peripheral autonomic nervous system is not controlled by the upper cognition of the brain. The peripheral autonomic nervous system regulates physiological functions such as respiration, heart rate and skin conductance, so electrocardiography (ECG) [24] and electrodermal activity (EDA) [25] can be used to measure the fatigue degree and emotional changes of users. There is also an eye measurements method that evaluates QoE by measuring eye gaze tracking, blinking, or pupillometry [26].

EEG is one of the basic theoretical research methods for brain science. Human mental and physical activities are dependent on bioelectricity. The brain produces and transmits different but regular electrical signals all the time. Therefore, the physiological signals of brain activity can overcome the influence of user fatigue, preference, educational background and external environment when analyzing the user's real feelings [27]. When neurons in the brain fire, they penetrate the brain's dura and skull, creating a weak wave of electrical potential on the brain's skin. This allows non-invasive EEG measurements to infer the firing of intracranial neurons, which can be observed and collected by attaching special electrodes to the surface of the scalp [27]. The location of these electrodes is usually specified in the 10–20 standard system, and the appropriate reference electrode is selected. A standard system facilitates the spatial localization and signal tracking of electrodes in EEG signal analysis.

Induced event-related potentials (ERPs) [28], time-frequency domain analysis [29] and spatial brain connectivity [30] are important methods for EEG experiments and signal processing. ERPs is a special brain potential evoked by sensory stimulation and cognitive process in the brain. The relative strength of the component is significantly improved during the superposition averaging process. After the occurrence of sensory stimulation events, the waveforms of specific channel signals show distinct multiple fluctuations in sequence, and these peaks and troughs represent different patterns of ERPs. The middle-latency response generally refers to the potential induced by 50–200 ms, mainly including N100, P100, N200 and P200. In the paper [31], the authors pointed out that N100 was widely present in a variety of cognitive processing functions, including auditory, visual, behavioral and cognitive tasks, and it can reflect early simple sensory processing and can be used as a biomarker of neuroplasticity. P300 is the neural activity triggered by task-related target stimulus, which is an important aspect of ERPs research. It is a widely existing component that can be recorded and observed in the scalp, with a large amplitude and a wide span [32]. The P3a subcomponent reflects the top-down frontal attentional mechanism during task processing. Another subcomponent, P3b, reflects top-down temporoparietal activity related to memory mechanisms [33]. N400 can be used as a neurophysiological index for semantic priming, with the absolute value of N400 amplitude being smaller when a word is a good match with the previous word/context, and larger when the two do not match [34]. The time-frequency decomposition of non-stationary time signals, such as continuous wavelet transform (CWT) [35], discrete wavelet transform (DWT) [29] and empirical mode decomposition (EMD) [36], are effective EEG signal analysis methods, which can accurately capture and locate transient features in the time domain and the frequency domain to better understand the dynamic characteristics of the human brain. Assessing information exchange between brain regions is also a common method for analyzing EEG signals. This method can be combined with graph theory to analyze and quantify the structure, function and causality of the brain. The directed transfer function of the autoregressive model framework was proposed and used to determine the direction and frequency content of brain activity, and the validity of the DTF algorithm was verified by real neurobiological data [37,38]. In the paper [39], the authors validated a connection-based EEG feature detection method using ML based on tone-mapped high dynamic range videos and confirmed that DTF outperformed undirected functions.

It is clear from a large amount of research that visual stimuli have been studied far more than auditory stimuli. In the paper [40], the authors pointed out that there were not as many physiological studies on hearing as vision, so early auditory perception activation could be explored by means of physiological measurement and computational intelligence. In our previous article, we carried out some preliminary research, including recruiting volunteers, collecting EEG signal samples, selecting appropriate threshold of DTF to construct edge sets and using weighted degree for clustering [41]. The work of this paper was based on the previous work, so part of the previous experimental results are presented in Section 5.3.

## 3. Design of Experiments

### *3.1. Procedure*

The experiments were performed in the Wireless Multimedia Communication Lab (WMC) at Tsinghua University. The subjects were required to complete all the experimental contents in the professional EEG shielding room, as shown in Figure 2. The process of signal acquisition required the subject to complete all experiments in a professional EEG shielding chamber. This shielding room can strictly control external noise, indoor temperature, light, and electromagnetic interference. Mobile phones and other devices were banned during the experimental phase. Before the experiment, every participant was asked to read and sign an informed consent form. The researchers explained the experimental procedure and operation to the subjects in detail. The subjects did not know the specific principles and methods of the experiment. During the experiment, the subjects had to complete their tasks alone in the shielding room. Researchers could watch the indoor situation through a monitor in the control room and the brain waves of the subjects through a computer screen in real time. In special cases, researchers could communicate with the subject through the internal microphone and sound system as necessary.

We recruited 12 students and young teachers as volunteers, consisting of 6 females and 6 males, aged between 18 and 28. None of them had major illnesses. They all had normal hearing and had never had any neurological problems. Participants were tested in a soundproof, standardized EEG lab and asked to minimize blinking, make body movement, and swallow during the experiment. Two of the subjects' data were discarded due to the too many behavioral interference signals. We finally admitted EEG data from a total of 10 subjects [41].

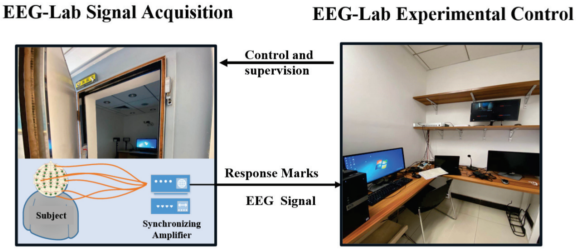


**Figure 2.** EEG experiment environment

### *3.2. Stimuli and Experimental Procedure*

In the experiment, four kinds of specially processed audio materials with very different semantic content were played through the headset, and each audio clip was played for 15 s. The four semantic contents were classical piano music, ocean waves, fire alarms and mosquitoes, all with periodic rhythms. Six levels of white Gaussian noise were added to each audio clip. The six Gaussian noise levels were defined according to the power spectral density of noise, which was 0, 0.001, 0.005, 0.01, 0.03 and 0.1. Depending on the level, the noise was added to the audio from 2s to 6s and lasted for 5 s. The noise of level 1 started from the second second; the noise of level 2 started from the third second, and so on. In the end, 24 different audio clips were obtained.

In each section of the experiment, the audio clips (24 in total) were randomly played twice. So, in the whole experiment, all the audio clips (24 in total) were played six times. After each audio clip was played, the subjects were asked whether they could tolerate the noise in the audio. A response of Y meant yes, and N meant no. At the end of each section, subjects rested for 3 min. At the end of the experiment, the subjects were asked to complete a subjective audio semantic questionnaire. We used the semantic difference method to make the subjects perform multiple perceptual evaluations on four different kinds of audio. The subjects were asked to evaluate three contrasting pairs of attributes. They were pleasant–unpleasant pair, relaxed–tense pair, and calm–upset pair. Matlab was used for audio material synthesis and signal processing, and Presentation, a program used for stimulation presentation and experimental control in physiological experiments, was used for stimulus materials. The whole experimental procedure is shown in Figure 3.

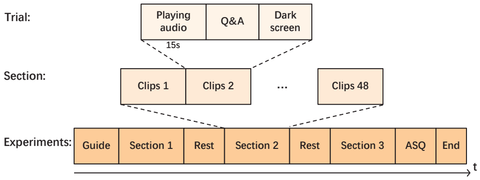


**Figure 3.** The experimental procedure consisted of three sections and two rests. In each session, 48 stimuli clips were played randomly.

## 4. Signal Processing

### *4.1. Directed Transfer Function*

In brain network research, directional functional brain connections can also be called causal brain connections. The information between the connected nodes is statistically causal. Methods for constructing causal connections mainly include directional transfer function (DTF) and partial directed coherence (PDC), and network connection thresholds need to be further selected for quantification. In this paper, we used the DTF method to construct the brain network and carried out degree feature extraction.

DTF is an autoregressive (AR) model [37], which can be described as

$$\sum_{d=0}^{D} A_d x_{t-d} = e_t$$

where $D$ is the model order determined by Akaike information criterion, $A_d$ is the delay matrix in AR model, and when $d = 0$, it is an identity matrix. $x_t = (x_{1,t}, x_{2,t}, \ldots, x_{k,t})$ is the the EEG data based on time series and $e_t = (e_{1,t}, \ldots, e_{k,t})$ is the vector of uncorrelated zero-means Gaussian white noise processes. If $x_{k,t}$is a stationary stochastic process, $A_d$ can be obtained according to the Yule–Walker equation. Then, the Z transformation gives the following result.

$$X(f) = H(f)E(f)$$

where $H(f)$ is the transfer function, $X(f)$ and $E(f)$ represent the transformed EEG data and noise data at frequency $f$. The DTF value (denoted by $DTF_{i,j}(f)$) is obtained by performing column square sum normalization by $H$ and indicates the information flow intensity between the $i$-th and $j$-th electrode.

There is a large amount of redundancy in the DTF coefficients. In the simulated signal test, only dimensions of 3, 4, 5 or 7 are used frequently [37,42], while in actual multichannel EEG signal processing, the dimensions are generally much greater than those in the simulated test. Therefore, we first simulated and tested the same dimensional vector time series system of the DTF algorithm so as to determine an appropriate threshold to construct the brain-connected network. We controlled the spectral radius $\rho(A_d)$ to solve the problem of randomly generating a large number of $A_d$ matrices while maintaining system stability in high-dimensional vector time series system simulation [37,41]. The formula is as follows.

$$r(A_d) \leq \rho(A_d) \leq R(A_d)$$

where $r(A_d)$ and $R(A_d)$ are the minimum and maximum row summation of $A_d$, respectively. In the process of simulation, we let each row summation of $A_d$ be a random variable obeying uniform distribution with extreme values of 0.30 and 0.95; thus, we had for all $i$.

$$\sum_{j=1}^{31} A_d(i,j) \sim U(0.30, 0.95)$$

Specifically, we gave the row summation and then randomly divided it into 5–16 parts as the elements of the corresponding line, indicating that $A_d$ was non-negative and $R(A_d) < 1$.

In our previous work [41], we found a strong correlation between the information flow accuracy of the DTF algorithm and the $A_d$ of the actual AR model through large-scale testing of random analog signals. Previous experimental results have shown that when 10% was chosen as the threshold for constructing the brain connectivity network, the accuracy of effective connectivity could be guaranteed at most densities of $A_d$.

### *4.2. Network Structure and Comprehensive Weighted Degree*

In order to characterize the intensity of information flow in the cerebral cortex, we constructed a brain connectivity graph by $DTF(f)$ denoted by $G_f^q = (V, A, W)$, where $V = \{1, 2, \ldots, 31\}$ is the vertex of the network, corresponding to 31 electrodes. $A = \{(i,j) | i, j \in 1, \ldots, 31 \text{ } and \text{ } i \neq j\}$ is the directed edge set of the graph and $W : A \to [0, 1]$ represents the weight of each directed edge. Figure 4 shows different brain connection networks constructed by a subject when listening to piano music of different quality levels. Different colors represent different connection strengths. As can be seen from the figure, the strength of noise in audio affected brain connectivity.

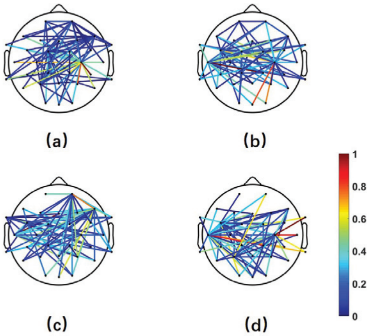


**Figure 4.** The brain connection networks of a subject when listening to piano music with 0 (**a**), 1 (**b**), 4 (**c**) and 5 (**d**) noise level.

To further quantify the information strength feature, for each vertex $v \in V(G)$, we calculated the following parameters.

$$deg(v) = \sum_{w \in ON(v) \setminus IN(v)} W(v,w) + \sum_{w \in IN(v) \setminus ON(v)} W(w,v) + \sum_{w \in ON(v) \cap IN(v)} max(W(v,w), W(w,v))$$

where $IN(v)$ and $ON(v)$ are the input and output neighbor of vertex $v$, respectively, and $deg(v)$ is the comprehensive weighted degree of $v$ , we also let $deg_{G_f^q}(V)$ denote the comprehensive weighted degree sequence of graph $G_f^q$, and $\lambda^q$ denote that of full-frequency band [41].

$$\lambda^q = \frac{\sum_f deg_{G_f^q}(V)}{f_{max} - f_{min} + 1}$$

Figure 5 shows the brain topography of comprehensive weighted degree of a user in four different audio scenarios under two extreme conditions (the audio with no noise and the audio with noise intensity of 0.1). It can be seen that the user's EEG response varies greatly under different conditions.

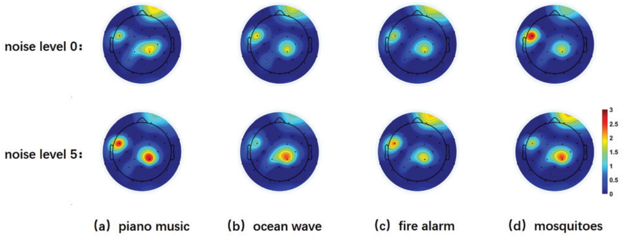


**Figure 5.** The brain topography of comprehensive weighted degree in four different audio scenarios under two extreme conditions.

### *4.3. Clustering*

For each given audio semantic scenario, we performed the clustering algorithm separately on $\lambda^0, \ldots \lambda^5$. Clustering optimization was carried out according to the error sum of squares criterion function.

$$J = \sum_{i=1}^{K} \sum_{j=1}^{N} w_{ji} \|\lambda^q - C_i\|$$

where $w$ is the membership coefficient, which is either zero or one. $\lambda^q$ is the feature data of K-means clustering. This comprehensive weighted degree was 31 dimensions. The clustering category was defined as the acceptable level space and the unacceptable level space, and the user's tolerance level in different audio semantics was determined by the clustering sample subordination, which was defined as the proportion of EEG signal samples classified into the unacceptable level category at different noise levels.

## 5. Result and Discussion

### *5.1. Results of Subjective Data Analysis on Noise Level*

Figure 6 shows the statistical subjective evaluation results of the number of times the user experiences noise that affects audio quality. It can be seen from the results that the pure subjective evaluation of users was not completely consistent with the objective facts. In many cases, the subjective evaluation results were intuitive but not reliable. For example, in the case of the sound of ocean waves, when the noise level was low, there was no negative evaluation. Users did not make a lot of negative quality evaluations, even when the noise level reached level 4, which was unexpected. In addition, although subjects were required to evaluate only the impact of noise on audio quality, in the last two audio scenarios of the experiment, when the noise level was zero, a lot of negative evaluations on audio quality had been received. In fact, the objective audio quality was very good at that point and did not include noise. This is the disadvantage of subjective evaluation, which is the uncontrollable subjective arbitrariness of users.

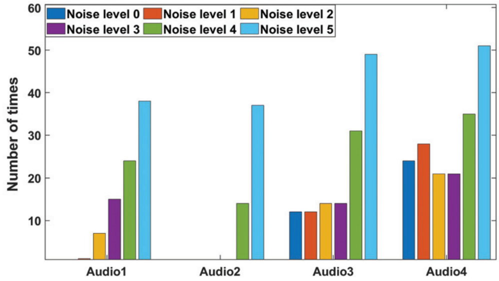


**Figure 6.** Number of times the user experiences noise that affects audio quality.

### 5.2. Results of Semantic Questionnaire Analysis

The attributes and semantic questionnaire analysis results are shown in Figure 7. It can be clearly seen from the figure that the perceptive semantic radar map of the four audio scenarios expressed two completely different audio emotions. This data and result can also be seen in our previous work [41]. The details were discussed in Section 5.3, together with the physiological data results.

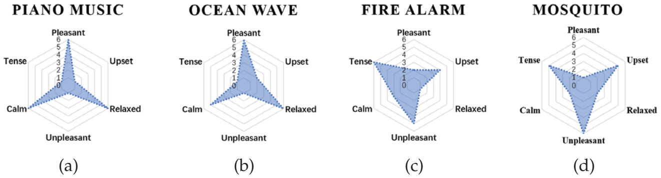


**Figure 7.** The subjective audio semantic questionnaire: the result of multiple perceptual evaluations on four different kinds of audio. (**a**) Piano music (**b**) Ocean wave (**c**) Fire alarm (**d**) Mosquito.

### 5.3. Perceptual Tolerance

An important goal of our analysis of EEG signals is to find the level of noise perceptual tolerance, when the noise level is higher than the perceptual tolerance, almost all subjects would show an intolerable trend. According to general experience, the perceptual tolerance of humans to audio noise should be determined by the value of the SNR. Figure 8 shows the SNR results of all audio stimulus materials with noise in our experiment. As can be seen from the Figure 8, the value of SNR decreases significantly with the increase in noise level. In addition, due to the different semantics of the audio scene, the value of SNR with the same noise level fluctuates in a small range. However, the physiological signal analysis results given in Figure 9 show that humans have different perceptual tolerance for the same noise level. In this work, the brain map of the comprehensive weighting degree was very different from that of high-intensity audio when users listened to raw audio and low-intensity-noise audio. Therefore, the comprehensive weighting degree of the full frequency can be used as EEG features for the clustering algorithm. Figure 9 shows the clustering visualization results as block diagrams of all subjects.

It can be clearly seen from Figure 9 that the user's noise tolerance level for a particular audio scenario was determined. Specifically, the user's limits of audio 1, 2, 3 and 4 were noise levels 1, 2, 4 and 4, respectively. We suspected the above results were related to the audio scenario and the difference between the original audio signal and the noise. So, we focused on the analysis of the semantic environment of the audio and the absolute integral value of the deviation between the four semantic audios with different levels of white noise.

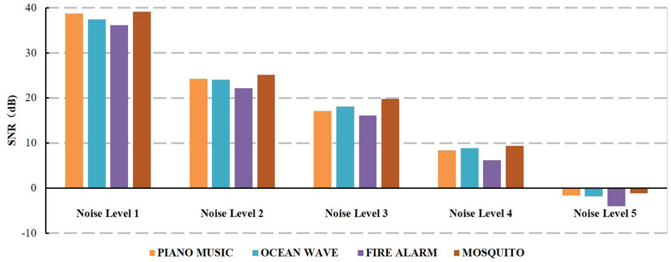


**Figure 8.** The SNR of audio stimulus materials.

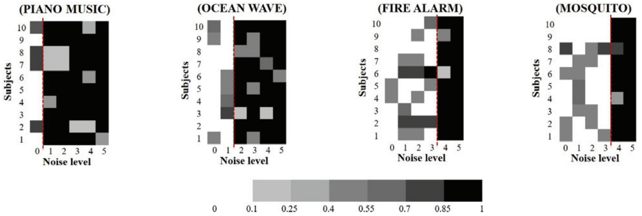


**Figure 9.** Clustering visualization results with comprehensive weighted degree based on DTF: A lighter block indicates a lower degree of subordination, and a deeper block indicates a higher degree of subordination. Red dashed lines represent the determination of clustering result.

Combined with the results of the perception semantic questionnaire results in Figure 7, it can be seen that based on the choices of all subjects, the smooth piano music and ocean waves make people feel pleasant, relaxed and calm. Under this situation, even the low-intensity white Gaussian noise on such audio will have a great influence on the subject's quality of experience; the user will be very sensitive to the noise, and their brainwave signal will significantly change. A different situation appears in audio 3 and 4. The fire alarm makes subjects feel tense and unpleasant, and mosquito audio makes subjects more upset. In this semantic audio environment, the subject's sensitivity to noise is reduced, and the perceptual tolerance of noise intensity is increased. Different audio scenes bring different perceptual emotions to people, which perfectly explains that humans' perceptual tolerance does not exactly correspond to the objective SNR of the audio. Figure 10 gives more details about the signal absolute difference integral proportion difference between audio with five levels of noise and raw audio under four different audio scenarios. This is a strong explanation for the results that the perceptual tolerance of audio 3 and 4 are higher than that of audio 1 and 2. The clustering results can also be seen in our previous work [41].

In conclusion, the perceptual tolerance of human perception of noise was related to the audio semantic environment perceived by users, and it was inversely proportional to the signal absolute difference integral proportion difference between audio with noise and raw audio under different audio scenarios. Moreover, both EEG signals analysis and subjective evaluations indicated that users were more sensitive to noise-induced quality changes in the calming and soothing audio scenario.

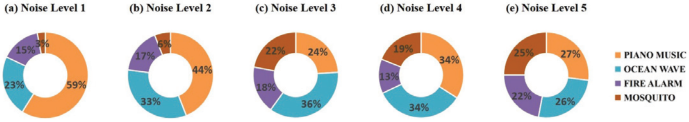


**Figure 10.** The signal absolute difference integral proportion difference between audio with five levels of noise and raw audio under four different audio scenarios.

### *5.4. Connectivity Analysis of Related Brain Regions*

To better illustrate the experimental result, we presented the comprehensive weighted degree of key channel signal of ten users with qualified experimental data. We defined the key channel as degree >1 in the audio condition (high-quality audio or audio with noise), and compared with level 0, the amplitude of level 5 increased by more than 10%. The specific values are shown in the table below.

The brain is divided into frontal, parietal, temporal, and occipital regions. The naming of the channel electrodes on the EEG cap is refined according to the location of the four brain regions. The channel F represents the frontal region, P represents the c region, T represents the temporal region, O represents the occipital region, C represents the central region, FC represents the frontal central region, CP represents the central parietal region, FP represents the frontal pole region, the singular represents the left brain, the even represents the right brain, and Z represents the middle region.

As can be seen from Table 1, when users heard the audio, the degree of nodes of CP related channels degree was higher than that of other nodes (8/10 users), and the degree of nodes of FC related channels degree was higher than that of other nodes (8/10 users), too, indicating that certain brain regions were activated after users heard the audio stimulation. We found that no matter the audio scenario, the value of node degree would increase significantly when there was noise, indicating that the activation degree of the electrical nerve signal in the brain area increased. For example, under four audio scenarios, the channel degree of the original audio and the audio with noise level 5 increased by 39.59%, 35.08%, 16.07%, and 41.66%, respectively. For another example, the CP2 channel degree of user 1 increased by 28.2%,65.92%, and 32.3% under the audio scenarios 1, 2 and 4. In audio scenario 3, the degree of channel CP5 increased by 13.99% in the same brain area. Similarly, the increase in FC-related channels was also obvious. Under audio scenarios 1, 2 and 3, the degree of FC2 of user 4 increased by 105.88%, 37.97%, and 28.78%, respectively. The degree of FC6 in the same area increased by 10.16% under audio scenario 2 and 18.75% under audio scenario 4. These statements suggested that noise had a greater effect on the brain regions where the channels mentioned above were located. In particular, the central parietal region where CP channels were located and the frontal central region where FC channels were located were cognitive-integration-related brain regions and preference-decision-related brain regions. These were consistent with previous research on brain perception [43]. All of these were consistent conclusions, regardless of the individual or the audio scenario. However, activation of the brain regions did not rule out individual differences. For example, when user 7 was under audio scenario 1 and scenario 4, the degree value and the increase in F3 and Fz channels in the frontal regions were both great.

**Table 1.** The values and ranges of degree.

| User | Audio Scene | Channel | The Values and Ranges of Degree |
|---|---|---|---|
| 1 | 1 | 23:CP2 | 1.95, 2.50, 28.2% |
| | 2 | 23:CP2 | 1.79, 2.97, 65.92% |
| | 3 | 11:CP5 | 2.43, 2.77, 13.99% |
| | 4 | 6:FC5 | 0.95, 1.18, 24.21% |
| | | 23:Cp2 | 1.95, 2.58, 32, 3% |
| 2 | 1 | 23:CP2 | 2.02, 2.50, 23.76% |
| | 2 | 23:CP2 | 1.79, 2.97, 65.92% |
| | 3 | 11:CP5 | 2.42, 2.86, 18.18% |
| | 4 | 6:CP5 | 0.89, 1.18, 32.58% |
| | | 23:CP2 | 1.96, 2.58, 31.63% |

**Table 1.** *Cont.*

| User | Audio Scene | Channel | The Values and Ranges of Degree |
|---|---|---|---|
| 3 | 1 | 28:FC6 | 1.07, 1.33, 24.29% |
| | 2 | 23:CP2 | 1.74, 1.79, 2.87% |
| | | 24:Cz | 1.82, 2.11, 15.93% |
| | | 19:P4 | 2.13, 2.60, 22.06% |
| | 3 | 28:FC6 | 0.83, 1.26, 51.8% |
| | 4 | 23:CP2 | 1.39, 1.53, 10.07% |
| | | 19:P4 | 2.31, 2.56, 10.82% |
| 4 | 1 | 29:FC2 | 0.85, 1.75, 105.88% |
| | 2 | 28:FC6 | 3.05, 3.36, 10.16% |
| | | 29:FC2 | 0.79, 1.09, 37.97% |
| | 3 | 24:Cz | 2.77, 3.16, 14.07% |
| | | 29:FC2 | 0.66, 0.85, 28.78% |
| | 4 | 28:FC6 | 3.04, 3.61, 18.75% |
| 5 | 1 | 6:FC5 | 1.66, 2.60, 56.62% |
| | | 23:CP2 | 1.97, 2.75, 39.59% |
| | 2 | 12:CP1 | 0.36, 1.18, 227.77% |
| | | 23:CP2 | 1.71, 2.31, 35.08% |
| | 3 | 6:FC5 | 1.64, 2.27, 38.41% |
| | | 23:CP2 | 1.68, 1.95, 16.07% |
| | | 31:F8 | 1.34, 1.65, 23.13% |
| | 4 | 23:CP2 | 1.68, 2.38, 41.66% |
| | | 31:F8 | 1.16, 1.59, 37.06% |
| 6 | 1 | 6:FC5 | 1.75, 2.24, 28.00% |
| | 2 | 6:FC5 | 1.45, 2.28, 57.24% |
| | | 7:FC1 | 0.84, 1.30, 54.76% |
| | | 8:C3 | 1.56, 2.05, 31.41% |
| | 3 | 7:FC1 | 0.64, 0.84, 31.25% |
| | 4 | 6:FC5 | 1.64, 2.58, 57.31% |
| 7 | 1 | 2:Fz | 1.04, 1.23, 18.26% |
| | | 14:P3 | 0.93, 1.06, 13.97% |
| | 2 | 12:CP1 | 1.52, 1.73, 13.81% |
| | | 14:P3 | 1.15, 1.63, 41.73% |
| | 3 | 12:CP1 | 1.04, 1.51, 45.19% |
| | | 22:CP6 | 1.11, 1.38, 24.32% |
| | | 28:FC6 | 0.73, 1.05, 43.83% |
| | 4 | 2:Fz | 1.13, 1.48, 30.97% |
| | | 14:P3 | 1.06, 1.39, 31.13% |
| 8 | 1 | 13:PZ | 0.98, 1.22, 24.48% |
| | | 23:CP2 | 0.52, 1.26, 142.3% |
| | 2 | 12:CP1 | 1.42, 1.77, 24.64% |
| | 3 | 23:CP2 | 0.55, 1.25, 127.27% |
| | 4 | 29:FC2 | 0.57, 0.69, 21.05% |
| 9 | 1 | 12:CP1 | 0.54, 1.04, 92.59% |
| | | 23:CP2 | 2.2, 2.42, 10.00% |
| | 2 | 11:CP5 | 1.78, 2.29, 28.65% |
| | 3 | 12:CP1 | 0.72, 1.03, 43.05% |
| | 4 | 23:CP2 | 2.36, 2.66, 12.71% |
| 10 | 1 | 6:FC5 | 1.64, 1.77, 7.92% |
| | 2 | 29:FC2 | 0.61, 0.68, 11.47% |
| | 3 | 11:CP5 | 1.78, 1.86, 4.49% |
| | 4 | 6:FC5 | 1.26, 1.82, 44.44% |

## 6. Conclusions

This paper discussed the evaluation methods of human subjective perception from two aspects. They were the analysis of the physiological signals from the central nervous system and the users' subjective behavioral data. The EEG was used to record real brain wave data, and brain connectivity maps were constructed to obtain the perceptual tolerance degree of audio noise in different scenarios. The relationship between audio signal-to-noise ratio, audio scenarios, user emotions and noise perception tolerance was analyzed comprehensively. Meanwhile, a change in brain activity intensity was also demonstrated.

**Author Contributions:** Conceptualization, B.G.; Methodology, B.G. and K.L.; Software, K.L.; Supervision, Y.D.; Writing—original draft, B.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Fundamental Research Funds for the Central Universities(CUC220C007, CUC22GZ007).

**Institutional Review Board Statement:** The data collection part of the study was conducted at Tsinghua University, and this study was approved by the Medical Ethics Committee of Tsinghua University (1100000118937).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** This work was also supported by Communication University of China and Tsinghua University-China Mobile Communications Group Co., Ltd. Joint Institute. We wish to thank Tsinghua WMC EEG Lab for providing experimental conditions. A small part of this work was presented at the conference IWCMC, and we have officially obtained IEEE permission to reuse the materials.

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

## Abbreviations

| | |
|---|---|
| QoE | Quality of Experience. |
| EEG | Electroencephalogram. |
| QoS | Quality of Service. |
| MOS | Mean Opinion Score. |
| ML | Machine learning. |
| DL | Deep learning. |
| DTF | Directional Transfer Function. |
| ERPs | Event-Related Potentials. |
| NIRS | Near-Infrared Spectroscopy. |
| fMRI | Functional Magnetic Resonance Imaging. |
| MEG | Magnetoencephalography. |
| ECG | Electrocardiography. |
| EDA | Electrodermal Activity. |
| CWT | Continuous Wavelet Transform. |
| DWT | Discrete Wavelet Transform. |
| EMD | Empirical Mode Decomposition. |
| AR | Autoregressive. |
| PDC | Partial Directed Coherence. |
| SNR | Signal to Noise Ratio. |

## References

1. Wu, Y.; Li, J.; Yuan, Y.; Qin, A.K.; Gong, M.G. Commonality Autoencoder: Learning Common Features for Change Detection From Heterogeneous Images. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 4257–4270. [CrossRef] [PubMed]
2. Wu, Y.; Mu, G.; Qin, C.; Miao, Q.; Zhang, X. Semi-Supervised Hyperspectral Image Classification via Spatial-Regulated Self-Training. *Remote Sens.* **2020**, *12*, 159. [CrossRef]
3. Moldovan, A.; Ghergulescu, I.; Weibelzahl, S.; Muntean, C.H. User-centered EEG-based multimedia quality assessment. In Proceedings of the 2013 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), London, UK, 5–7 June 2013; pp. 1–8.
4. Wu, Y.; Zhang, L.; Lv, T.; Guo, R.; Xing, L.; Wang, Y. An Intelligent Perception Model and Parameters Adjust Method for Quality of Experience. *Electronics* **2022**, *11*, 1732. [CrossRef]

5. Ahmad, A.; Mansoor, A.B.; Barakabitze, A.A.; Hines, A.; Atzori, L.; Walshe, R. Supervised-learning-Based QoE Prediction of Video Streaming in Future Networks: A Tutorial with Comparative Study. *IEEE Commun. Mag. Artic. News Events Interest Commun. Eng.* **2021**, *59*, 88–94. [CrossRef]
6. Zhang, Q.; Zhu, W.; Zhang, Y.Q. End-to-End QoS for Video Delivery Over Wireless Internet. *Proc. IEEE* **2005**, *93*, 123–134. [CrossRef]
7. Skorin-Kapov, L.; Varela, M.; Hobfeld, T.; Chen, K.T. A Survey of Emerging Concepts and Challenges for QoE Management of Multimedia Services. *Acm Trans. Multimed. Comput. Commun. Appl.* **2018**, *14*, 1–29. [CrossRef]
8. ITU-T. 1865. Available online: https://www.itu.int/en/ITU-T/Pages/default.aspx (accessed on 7 September 2022).
9. New Appendix I-Definition of Quality of Experience (QoE). ITU-T Rec. P.10/G.100 Appendix 1. 2007. Available online: https://cir.nii.ac.jp/crid/1570291225912681600 (accessed on 14 July 2008).
10. Song, J.; Yang, F.; Zhou, Y.; Wan, S.; Wu, H.R. QoE Evaluation of Multimedia Services Based on Audiovisual Quality and User Interest. *IEEE Trans. Multimed.* **2016**, *18*, 444–457. [CrossRef]
11. Yang, M.; Wang, S.; Calheiros, R.N.; Yang, F. Survey on QoE Assessment Approach for Network Service. *IEEE Access* **2018**, *6*, 48374–48390. [CrossRef]
12. Mok, R.K.P.; Luo, X.; Chan, E.W.W.; Chang, R.K.C. QDASH: A QoE-aware DASH system. In Proceedings of the Proceedings of the Third Annual ACM SIGMM Conference on Multimedia Systems, Chapel Hill, NC, USA, 22–24 February 2012.
13. Wang, Y.; Agarwal, M.; Lan, T.; Aggarwal, V. Learning-Based Online QoE Optimization in Multi-Agent Video Streaming. *Algorithms* **2022**, *15*, 227. [CrossRef]
14. Cassani, R.; Moinnereau, M.A.; Falk, T.H. A Neurophysiological Sensor-Equipped Head-Mounted Display for Instrumental QoE Assessment of Immersive Multimedia. In Proceedings of the 2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX), Cagliari, Italy, 29 May–1 June 2018; pp. 1–6.
15. Machado, V.A.; Silva, C.N.; Oliveira, R.S.; Melo, A.M.; Hirata, C.M. A new proposal to provide estimation of QoS and QoE over WiMAX networks: An approach based on computational intelligence and discrete-event simulation. In Proceedings of the 2011 IEEE Latin-American Conference on Communications (LATINCOM), Belem, Brazil, 24–26 October 2011.
16. Huang, R.; Xin, W.; Lv, C.; Li, X.; Zhang, S. Prediction Model for User's QoE in Imbalanced Dataset. In Proceedings of the 2015 First International Conference on Computational Intelligence Theory, Systems and Applications (CCITSA), Ilan, Taiwan, 10–12 December 2015.
17. Deressa, M.; Sheng, M.; Wimmers, M.; Liu, J.; Mekonnen, M. Maximizing Quality of Experience in Device-to-Device Communication Using an Evolutionary Algorithm Based on Users' Behavior. *IEEE Access* **2017**, *5*, 3878–3888. [CrossRef]
18. Zhang, H.; Hu, H.; Gao, G.; Wen, Y.; Guan, K. DeepQoE: A unified Framework for Learning to Predict Video QoE. In Proceedings of the IEEE International Conference on Multimedia & Expo, San Diego, CA, USA, 23–27 July 2018; pp. 1–6.
19. Kwon, M.; Cho, H.; Won, K.; Ahn, M.; Jun, S.C. Use of Both Eyes-Open and Eyes-Closed Resting States May Yield a More Robust Predictor of Motor Imagery BCI Performance. *Electronics* **2020**, *9*, 690. [CrossRef]
20. Spence, S. The Cognitive Neurosciences. *J. Cogn. Neuroence* **1995**, *7*, 514. [CrossRef]
21. Laghari, K.R.; Gupta, R.; Arndt, S.; Antons, J.; Schleicher, R.; Möller, S.; Falk, T.H. Neurophysiological experimental facility for Quality of Experience (QoE) assessment. In Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), Ghent, Belgium, 27–31 May 2013; pp. 1300–1305.
22. Kim, D.; Yong, J.J.; Kim, E.; Yong, M.R.; Park, H.W. Human brain response to visual fatigue caused by stereoscopic depth perception. In Proceedings of the 2011 17th International Conference on Digital Signal Processing (DSP), Corfu, Greece, 6–8 July 2011; pp. 1–5.
23. Miettinen, I.; Tiitinen, H.; Alku, P.; May, P.J. Sensitivity of the human auditory cortex to acoustic degradation of speech and non-speech sounds. *BMC Neurosci.* **2010**, *11*, 24. [CrossRef]
24. Kroupi, E.; Hanhart, P.; Lee, J.S.; Rerabek, M.; Ebrahimi, T. Predicting subjective sensation of reality during multimedia consumption based on EEG and peripheral physiological signals. In Proceedings of the IEEE International Conference on Multimedia & Expo, Chengdu, China, 14–18 July 2014; pp. 1–6.
25. Keighrey, C.; Flynn, R.; Murray, S.; Murray, N. A Physiology-based QoE Comparison of Interactive Augmented Reality, Virtual Reality and Tablet-based Applications. *IEEE Trans. Multimed.* **2021**, *23*, 333–341. [CrossRef]
26. Liu, H.; Heynderickx, I. Visual Attention in Objective Image Quality Assessment: Based on Eye-Tracking Data. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 971–982.
27. Moon, S.E.; Lee, J.S. Implicit Analysis of Perceptual Multimedia Experience Based on Physiological Response: A Review. *IEEE Trans. Multimed.* **2017**, *19*, 340–353. [CrossRef]
28. Liu, X.; Tao, X.; Xu, M.; Zhan, Y.; Lu, J. An EEG-Based Study on Perception of Video Distortion Under Various Content Motion Conditions. *IEEE Trans. Multimed.* **2020**, *22*, 949–960. [CrossRef]
29. Adeli, H.; Zhou, Z.; Dadmehr, N. Analysis of EEG records in an epileptic patient using wavelet transform. *J. Neurosci. Methods* **2003**, *123*, 69–87. [CrossRef]
30. Friston, K.J. Functional and effective connectivity: A review. *Brain Connect* **2011**, *1*, 13–36. [CrossRef]
31. Joseph, G.H.; Michelle, B.E.; Eugene, D.; Seidman, L.J.; Sarah, G.; April, K.; Woodberry, K.A.; Ashley, R.; Sahil, T.; Kyle, O. N100 Repetition Suppression Indexes Neuroplastic Defects in Clinical High Risk and Psychotic Youth. *Neural Plast.* **2016**, *2016*, 4209831.

32. Bachiller, A.; Lubeiro, A.; Díez, Á; Suazo, V.; Domínguez, C.; Blanco, J.A.; Ayuso, M.; Hornero, R.; Poza, J.; Molina, V. Decreased entropy modulation of EEG response to novelty and relevance in schizophrenia during a P300 task. *Eur. Arch. Psychiatry Clin. Neurosci.* **2015**, *265*, 525–535. [CrossRef] [PubMed]
33. Polich, J. Updating P300: An integrative theory of P3a and P3b. *Clin. Neurophysiol.* **2007**, *118*, 2128–2148. [PubMed]
34. Boyd, J.E.; Patriciu, I.; McKinnon, M.C.; Kiang, M. Test-retest reliability of N400 event-related brain potential measures in a word-pair semantic priming paradigm in patients with schizophrenia. *Schizophr. Res.* **2014**, 158, 195. [CrossRef]
35. Cohen, M.X. A better way to define and describe Morlet wavelets for time-frequency analysis. *NeuroImage* **2019**, *199*, 81–86. [CrossRef] [PubMed]
36. Sweeney-Reed, C.M.; Nasuto, S.J. A novel approach to the detection of synchronisation in EEG based on empirical mode decomposition. *J. Comput. Neurosci.* **2007**, *23*, 79–111. [CrossRef]
37. Kaminski, M.J.; Blinowska, K.J. A new method of the description of the information flow in the brain structures. *Biol. Cybern.* **1991**, *65*, 203–210. [CrossRef]
38. Van Wijk, B.C.; Stam, C.J.; Daffertshofer, A. Comparing brain networks of different size and connectivity density using graph theory. *PLoS ONE* **2010**, *5*, e13701. [CrossRef]
39. Moon, S.E.; Lee, J.S. EEG Connectivity Analysis in Perception of Tone-mapped High Dynamic Range Videos. In Proceedings of the 23rd ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015; pp. 987–990.
40. Tian, X.; Ding, N.; Teng, X.; Bai, F.; Poeppel, D. Imagined speech influences perceived loudness of sound. *Nat. Hum. Behav.* **2018**, *2*, 225–234. [CrossRef]
41. Geng, B.; Liu, K.; Duan, Y.; Song, Q.; Shi, J. A Novel EEG Based Directed Transfer Function for Investigating Human Perception to Audio Noise. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 923–928.
42. Baccal, L.A.; Sameshima, K. Partial directed coherence: A new concept in neural structure determination. *Biol. Cybern.* **2001**, *84*, 463–474. [CrossRef]
43. Wang, R.W.; Chang, Y.C.; Chuang, S.W. EEG spectral dynamics of video commercials: Impact of the narrative on the branding product preference. *Sci. Rep.* **2016**, *6*, 36487. [CrossRef]

electronics

MDPI

*Article*

# An Improved Nonlinear Tuna Swarm Optimization Algorithm Based on Circle Chaos Map and Levy Flight Operator

**Wentao Wang and Jun Tian ***

College of Software, Nankai University, Tianjin 300071, China
* Correspondence: jtian@nankai.edu.cn

**Abstract:** The tuna swarm optimization algorithm (TSO) is a new heuristic algorithm proposed by observing the foraging behavior of tuna populations. The advantages of TSO are a simple structure and fewer parameters. Although TSO converges faster than some classical meta-heuristics algorithms, it can still be further accelerated. When TSO solves complex and challenging problems, it often easily falls into local optima. To overcome the above issue, this article proposed an improved nonlinear tuna swarm optimization algorithm based on Circle chaos map and levy flight operator (CLTSO). In order to compare it with some advanced heuristic algorithms, the performance of CLTSO is tested with unimodal functions, multimodal functions, and some CEC2014 benchmark functions. The test results of these benchmark functions are statistically analyzed using Wilcoxon, Friedman test, and MAE analysis. The experimental results and statistical analysis results indicate that CLTSO is more competitive than other advanced algorithms. Finally, this paper uses CLTSO to optimize a BP neural network in the field of artificial intelligence. A CLTSO-BP neural network model is proposed. Three popular datasets from the UCI Machine Learning and Intelligent System Center are selected to test the classification performance of the new model. The comparison result indicates that the new model has higher classification accuracy than the original BP model.

**Keywords:** artificial intelligence; circle chaotic map; Levy flight; nonlinear adaptive weight; tuna swarm optimization

**Citation:** Wang, W.; Tian, J. An Improved Nonlinear Tuna Swarm Optimization Algorithm Based on Circle Chaos Map and Levy Flight Operator. *Electronics* **2022**, *11*, 3678. https://doi.org/10.3390/electronics11223678

Academic Editor: Javid Taheri

Received: 23 October 2022
Accepted: 7 November 2022
Published: 10 November 2022

## 1. Introduction

Nowadays, many engineering problems in real life have become more and more complex and challenging. High-quality solutions can help people effectively reduce resource investment. Because most production practice problems are multivariate, nonlinear, and have many complex constraints, the traditional branch and bound algorithm [1], conjugate gradient method [2], and dynamic programming method [3] cannot achieve remarkable results regarding these problems. The meta-heuristic algorithm has the characteristics of strong global search ability, no dependence on gradient information, and wide adaptability. It can effectively overcome the shortcomings of traditional optimization algorithms. Much of the research on meta-heuristic algorithms has shown that these algorithms are able to solve nonlinear optimization problems [4,5]. Many researchers tend to use meta-heuristic algorithms to solve complex engineering problems. Now, meta-heuristic algorithms are applied in various fields, such as workshop scheduling [6], task optimization [7], engineering management [8–10], and others.

The meta-heuristic algorithm is a mathematical method inspired by biological behavior and some physical phenomena in nature. These methods are used to solve complex problems in real life [11]. The meta-heuristic algorithm has the advantages of a simple structure, fewer hyperparameters, and being easy to understand. Based on these advantages, it has become an important method for solving optimization problems today. Meta-heuristic algorithms can be divided into four categories: swarm intelligence algorithms [12], evolutionary algorithms [13], human-based algorithms [14], and physical and chemical-based algorithms [15]. The swarm intelligence algorithms simulate the behavior of animal populations. Each individual in the population is a candidate solution. They are randomly explored in the search space, which effectively avoids the possibility of entering the local optimum. Some classic and newly proposed swarm intelligence algorithms include Golden Jackal Optimization (GJO) [16], the Gray Wolf Optimization Algorithm (GWO) [17], and the Poplar Optimization Algorithm (POA) [18]. Some classical evolutionary algorithms include Genetic Algorithms [19] and the Biogeographic-Based Optimization Algorithm (BBO) [20], etc. Meta-heuristic algorithms can effectively enhance the efficiency of engineering practice. This has attracted more and more scholars' attention.

In many industrial problems, specific solution functions can be established with mathematical models. How to solve complex function optimization problems has become a focus of current research. For the optimization problems with fewer constraints and dimensions, the traditional mathematical methods can achieve outstanding results. Although meta-heuristic algorithms have very good performance in dealing with complex and high dimensional optimization problems, the convergence speed of simple meta-heuristic algorithms still needs to be improved. Sometimes with a single meta-heuristic algorithm, it is difficult to get rid of the attraction of local extremum. To further enhance the optimization capability of meta-heuristic algorithms, many experts try to use different strategies to improve them. Zhongzhou Du introduced Levy flight in the iterative process of PSO, which accelerated the optimization speed of PSO [21]. Hang Yu used a chaotic mapping strategy to improve the GWO initialization method, which improved the accuracy of the GWO solution [22]. Xiaoling Yuan introduced adaptive weight into the PSO algorithm, which greatly strengthened its global search capability [23]. So-Youn Park combined CS with oppositional learning, making the CS converge faster [24]. W. Xie used the golden sine operator to improve the Black Hole algorithm (BH) [25], giving it better exploration performance [26].

Xie et al. proposed a new meta-heuristic algorithm called the tuna swarm optimization algorithm (TSO) [27] in 2021 after observing the foraging behavior of tuna swarms. There are two common foraging strategies for tuna swarms: spiral foraging strategy and parabolic foraging strategy. TSO searches for the global optimal value by simulating the common individual in the tuna swarm to follow the optimal individual in the swarm to attack the prey. Comparing TSO with the Whale Optimization Algorithm (WOA) [28], the Salp Swarm Algorithm (SSA) [29], and some other advanced algorithms, the comparison results indicate that TSO outperforms the competitors. The tuna swarm optimization algorithm has the advantages of less parameters and easy realization. Therefore, after it was proposed, TSO has been widely studied and applied to engineering practice. Although TSO performs very well in many engineering practices, it still has some shortcomings. Firstly, TSO cannot efficiently search for the global optimal value. It is easily attracted by local extremum. Secondly, TSO does not converge fast enough. Finally, the followers of the optimal individual blindly follow the later. There is a lack of local exploitation. At present, Hu et al. have used Gaussian mutation to improve the TSO algorithm, and have applied the improved algorithm to photovoltaic power prediction [30]. Kumara et al. improved the TSO algorithm by using chaotic maps to increase the diversity of the algorithm population [31]. This paper proposes an improved tuna swarm optimization algorithm (CLTSO) based on the Circle chaotic map [32], Levy flight operator, and nonlinear adaptive operator. The innovations made in this article are summarized as follows:

(1) At the CLTSO initialization stage, this paper introduces the Circle chaotic map to uniformly generate individual positions. Because the initial positions of tuna individuals are randomly generated, the initial tuna individuals are likely to cluster together. In this paper, the emergence of the initial individual aggregation problem can be effectively solved by introducing the Circle chaotic map.

(2) In CLTSO, the optimal individual and its follower positions are updated by using Levy flight strategy. Because Levy flight uses a combination of long and short steps, it can significantly enlarge the search scope of CLTSO.

(3) In the iterative process of CLTSO, a nonlinear convergence factor is introduced to balance the exploration and the exploitation. In CLTSO, a large convergence factor in the initial iteration can bring the common individuals closer to the optimal individuals. A smaller convergence factor at the end of iteration increases the capability of followers to explore local scope.

This article covers the following aspects: Section 1 introduces some related content of the meta-heuristic algorithm and the tuna swarm optimization algorithm. Section 2 reviews the two foraging strategies of the original tuna swarm optimization algorithm. Section 3 introduces the improved Circle chaotic map strategy, the Levy flight operator, and the nonlinear adaptive weight operator, and the usage of these operators to improve TSO. Section 4 compares CLTSO with some classical and advanced meta-heuristics and makes some experimental analysis. Section 5 modifies the BP neural network based on CLTSO, and then tests the new model by using three popular datasets. Finally, Section 6 summarizes the content of the article.

The main mathematical symbols mentioned in this paper are shown in Table 1.

**Table 1.** Explanation of symbols.

| Symbol | Meaning |
|---|---|
| $X_i^{\text{int}}$ | Tuna individual in TSO |
| $ub$ | The upper boundary of the search space of TSO |
| $lb$ | The lower boundary of the search space of TSO |
| $NP$ | Population size of TSO |
| $\tau$ | Distance parameter |
| $\alpha_1$ | Weight parameters of tuna following the best individual |
| $\alpha_2$ | Weight parameters of tuna following the front individual |
| $p$ | Weight parameters in parabolic foraging strategy |
| $s$ | The step length of Levy flight |
| $t$ | Current number of iterations of the algorithm |
| $T_{Max}$ | Maximum number of iterations of the algorithm |
| $\alpha_{1i}$ | Improved version of $\alpha_1$ |
| $\alpha_{2i}$ | Improved version of $\alpha_2$ |
| $p_i$ | Improved version of $p$ |

## 2. An Overview of Tuna Optimization Algorithms

Tuna is the top predator in the ocean. Although tuna swim very fast, some small prey are more flexible than tuna. Therefore, in the process of predation, tuna often choose group cooperation to capture prey. The tuna swarm has two efficient predatory strategies, namely, the spiral foraging strategy and the parabolic foraging strategy. When the tuna swarm uses the parabolic foraging strategy, each tuna will follow the previous individual closely. The tuna swarm forms a parabola to surround the prey. When the tuna swarm adopts the spiral foraging strategy, the tuna swarm will aggregate into spiral shapes and drive prey to shallow water areas. Prey is more likely to be captured. By observing these two foraging behaviors of tuna swarm, researchers proposed a new swarm intelligence optimization called TSO.

### *2.1. Population Initialization*

There are $NP$ tunas in a tuna swarm. At the swarm initialization phase, the tuna swarm optimization algorithm randomly generates the initial swarm in the search space. The mathematical formulas for initializing tuna individuals are as follows:

$$\begin{aligned} X_i^{\text{int}} &= rand \cdot (ub - lb) + lb \\ &= \begin{bmatrix} x_i^1 & x_i^2 & \cdots & x_i^j \end{bmatrix} \\ &\begin{cases} i = 1, 2, \ldots, NP \\ j = 1, 2, \ldots, Dim \end{cases} \end{aligned} \tag{1}$$

where $X_i^{\text{int}}$ is the $i$-th tuna, $ub$ and $lb$ are the upper and lower boundaries of the range of tuna exploration, and $rand$ is a random variable with uniform distribution from 0 to 1. In particular, each individual, $X_i^{\text{int}}$, in the tuna swarm represents a candidate solution for TSO. Each individual tuna consists of a set of $Dim$-dimensional numbers.

### *2.2. Parabolic Foraging Strategy*

Herring and eel are the main food sources of tuna. When they encounter predators, they will use their speed advantage to constantly change their direction of swimming. It is very difficult for predators to catch them. Because tuna is less agile than their prey, the tuna swarm will take a cooperative approach to attack the prey. The tuna swarm will use the prey as a reference point to keep chasing prey. During predation, each tuna follows the previous individual, and the whole tuna swarm forms a parabola to surround the prey. In addition, the tuna swarm also uses a spiral foraging strategy. Assuming that the probability of the tuna swarm choosing either strategy is 50%, the mathematical model of parabolic foraging of the tuna swarm is as follows:

$$X_i^{t+1} = \begin{cases} X_{best}^t + rand \cdot (X_{best}^t - X_i^t) + TF \cdot p^2 \cdot (X_{best}^t - X_i^t), & \text{if rand } < 0.5 \\ TF \cdot p^2 \cdot X_i^t, & \text{if rand} \geq 0.5 \end{cases} \tag{2}$$

$$p = (1 - \frac{t}{t_{\max}})^{(t/t_{\max})} \tag{3}$$

where $t$ indicates that the $t$th iteration is currently running and $t_{\max}$ means the maximum number of iterations preset. $TF$ is a random value of 1 or $-1$.

### *2.3. Spiral Foraging Strategy*

Besides the parabolic foraging strategy, there is another efficient cooperative foraging strategy called the spiral foraging strategy. While chasing the prey, most tuna cannot choose the right direction, but a small number of tuna can guide the swarm to swim in the right direction. When a small group of tuna start chasing the prey, the nearby tuna will follow this small group of individuals. Eventually, the entire tuna swarm will form a spiral formation to catch the prey. When the tuna swarm adopts a spiral foraging strategy, individuals will exchange information with the best to follow individuals or adjacent individuals in the swarm. Sometimes the best individual is not able to lead the swarm to capture prey effectively. The tuna will then select a random individual in the swarm to follow. The mathematical formula of the spiral foraging strategy is as follows:

$$X_i^{t+1} = \begin{cases} \begin{array}{c} \alpha_1 \cdot (X_{rand}^t + \tau \cdot \left|X_{rand}^t - X_i^t\right| + \alpha_2 \cdot X_i^t), \\ i = 1 \\ \alpha_1 \cdot (X_{rand}^t + \tau \cdot \left|X_{rand}^t - X_i^t\right| + \alpha_2 \cdot X_{i-1}^t), \\ i = 2, 3, \ldots, NP \end{array} & , if \text{ rand} < \frac{t}{t_{\max}} \\ \begin{array}{c} \alpha_1 \cdot (X_{best}^t + \tau \cdot \left|X_{best}^t - X_i^t\right| + \alpha_2 \cdot X_i^t), \\ i = 1 \\ \alpha_1 \cdot (X_{best}^t + \tau \cdot \left|X_{best}^t - X_i^t\right| + \alpha_2 \cdot X_{i-1}^t), \\ i = 2, 3, \ldots, NP \end{array} & , if \text{ rand} \geq \frac{t}{t_{\max}} \end{cases} \tag{4}$$

where $X_i^{t+1}$ denotes the $i$-th tuna in the $t+1$ iteration. The current best individual is $X_{best}^t$. $X_{rand}^t$ is the reference point randomly selected in the tuna swarm. $\alpha_1$ is the trend weight coefficient to control the tuna individual swimming to the optimal individual or randomly selected adjacent individuals. $\alpha_2$ is the trend weight coefficient to control the tuna individual swimming to the individual in front of it. $\tau$ is the distance parameter that controls the distance between the tuna individual and the optimal individual or a randomly selected reference individual. Their mathematical calculation model is as follows:

$$\alpha_1 = a + (1 - a) \cdot \frac{t}{t_{\max}} \tag{5}$$

$$\alpha_2 = (1 - a) - (1 - a) \cdot \frac{t}{t_{\max}} \tag{6}$$

$$\tau = e^{bl} \cdot \cos(2\pi b) \tag{7}$$

$$l = e^{3\cos(((t_{\max}+1/t)-1)\pi)} \tag{8}$$

where $a$ is a constant to measure the degree of tuna following and $b$ is a random number uniformly distributed in the range of [0, 1].

### 2.4. Pseudocode of TSO

The pseudocode of the original TSO is displayed in Algorithm 1. The flow chart of TSO is displayed in Figure 1.

**Algorithm 1** Pseudocode of TSO Algorithm

```
Initialization: Set parameters $NP$, $Dim$, $a$, $z$ and $T_{\mathrm{Max}}$
Initialize the position of tuna $X_i$ (i = 1, 2, . . . , NP) by (1)
Counter $t$ = 0
while $T < T_{\mathrm{Max}}$ do
  Calculate the fitness value of all tuna
  Update the position and value of the best tuna $X^t_{best}$
  for (each tuna) do
      Update $\alpha_1$, $\alpha_2$, $p$ by (5), (6), (3)
        if ($rand < z$) then
              Update $X^{t+1}_i$ by (1)
      else if ($rand \geq z$) then
            if ($rand < 0.5$) then
                Update $X^{t+1}_i$ by (4)
            else if ($rand \geq 0.5$) then
                Update $X^{t+1}_i$ by (2)
t = t + 1
return the best fitness value f ($X_{best}$) and the best tuna $X_{best}$
```

In the iterative process of the TSO algorithm, each tuna will randomly choose to perform either the spiral foraging strategy or the parabolic foraging strategy. Tuna will also generate new individuals in the search range according to probability $Z$. Therefore, TSO will choose different strategies according to $Z$ when generating new individual positions. During the execution of the TSO algorithm, all tuna individuals in the population are constantly updated until the number of iterations reaches a predetermined value. Finally, the TSO algorithm returns the optimal individual in the population and its optimal value.

The following advantages of TSO can be seen from Algorithm 1: (1) The TSO algorithm has fewer adjustable parameters, which is beneficial to the implementation of the algorithm. (2) This algorithm will save the position of the best tuna individual in each iteration; even if the quality of the candidate solution decreases, it will not affect the location of the optimal value. (3) The TSO algorithm can keep the balance between exploitation and exploration by selecting two foraging strategies.

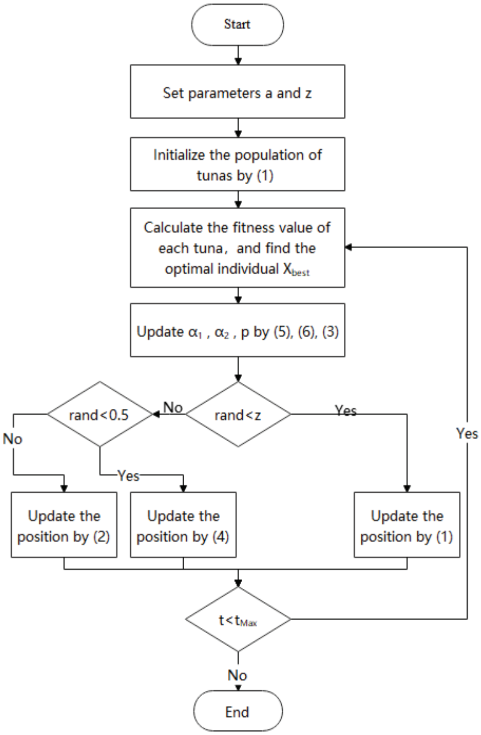

Start
Set parameters a and z
Initialize the population of tunas by (1)
Calculate the fitness value of each tuna，and find the optimal individual X<sub>best</sub>
Update α₁ , α₂ , p by (5), (6), (3)
rand<0.5
No
rand<z
Yes
No
Yes
Yes
Update the position by (2)
Update the position by (4)
Update the position by (1)
t<t<sub>Max</sub>
No
End

**Figure 1.** Flow chart of TSO.

## 3. The Improved Tuna Swarm Optimization Algorithm

This section introduces an improved nonlinear tuna swarm optimization algorithm, CLTSO, based on Circle chaotic map and Levy flight operator. Firstly, the population initialization using Circle chaotic map can increase the diversity of the swarm. The combination of TSO and Levy flight gives the algorithm an outstanding global exploration capability. Furthermore, a nonlinear adaptive weight operator is introduced to modify the weight coefficient of tuna following behavior in CLTSO. In CLTSO, the relationship between global exploration and local exploitation in the iterative process are well balanced.

### *3.1. Circle Chaotic Map*

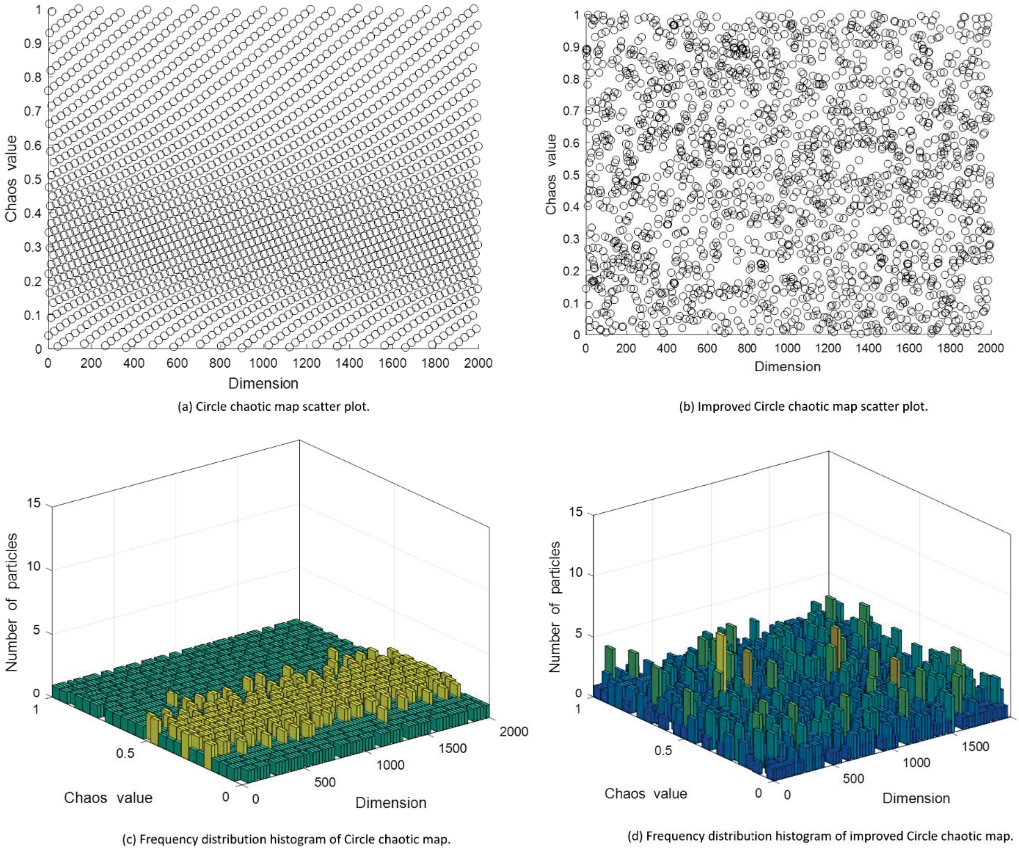
Many changes in nature are not random. They seem to conform to some special laws. Such a phenomenon is called chaos. Many movements in nature are chaotic [33]. Chaos is a random behavior, but it conforms to certain laws, which enables this operator to display more states in the search space of TSO [34].

Because the position of the tuna is randomly generated in the initialization phase of the tuna algorithm, it is easy to make the initial tuna gather at the same place. The initial tuna swarm does not fully cover the search space, resulting in a small difference between tuna individuals. This greatly reduces the global searching capability of the algorithm. The current popular chaotic mapping strategies are as follows: Tent [35], Logistic [36], Circle [37], Chebyshev [38], Sinusoidal [39], and Iterative chaotic map [40]. Studying the related literature on the above chaotic mapping strategies, we found that Circle chaotic map has a more stable chaotic value and has a higher coverage rate in the search space [41]. However, our experiments indicate that the distribution of Circle chaotic value is still not uniform. The chaotic values of the original Circle operator are clustered in the scope of [0.2, 0.5]. To make the chaotic value distribution more uniform, we improved the mathematical model of the Circle chaotic mapping strategy.

The mathematical modeling of the original Circle chaotic map is as follows:

$$x_{i+1} = \mathrm{mod}(x_i + 0.2 - (0.5/2\pi)\sin(2\pi x_i), 1) \tag{9}$$

where $x_i$ is the $i$th chaotic particle and $x_{i+1}$ is the ($i$ + 1)th chaotic particle. The scatter plot and frequency histogram of the initial candidate solution of the original Circle chaotic mapping operator are displayed in subgraphs (a) and (c) of Figure 2. In the Circle chaotic map experiment, the total number of particles is 2000. Chaotic particles denote the initial candidate solution of TSO.



(a) Circle chaotic map scatter plot.

(b) Improved Circle chaotic map scatter plot.

(c) Frequency distribution histogram of Circle chaotic map.

(d) Frequency distribution histogram of improved Circle chaotic map.

**Figure 2.** Frequency distribution histogram of improved Circle chaotic map.

As can be seen from subgraphs (a) and (c) of Figure 2, the chaotic particles are concentrated in the range of [0.2, 0.5] in the chaotic sequence initialized by Circle chaotic map. However, the initial candidate solutions are too concentrated, which will greatly reduce the population diversity of TSO. Therefore, the original Circle chaotic map is improved in this paper [42]. The mathematical modeling of the improved Circle chaotic map is as follows:

$$x_{i+1} = \mathrm{mod}(3.85x_i + 0.4 - (0.7/3.85\pi)\sin(3.85\pi x_i), 1) \tag{10}$$

where $x_i$ is the $i$th chaotic particle and $x_{i+1}$ is the ($i$ + 1)th chaotic particle.

The scatter plot and frequency histogram of the initial candidate solution of the improved Circle chaotic map operator are displayed in subgraphs (b) and (d) of Figure 2.

From (b) and (d), we can clearly see that, compared to the original Circle chaotic map, the particle distribution of the improved Circle chaotic map is more uniform. Each candidate solution particle of the algorithm is explored in the search space. Therefore, using the improved Circle chaotic map operator to modify TSO can obtain more uniform candidate solutions. The initial tuna individuals uniformly distributed in the search space of the algorithm can significantly increase the population diversity of TSO.

### *3.2. Levy Flight*

The movement and trajectory of many small animals and insects in life have the characteristics of Levy flight. These animals and insects include ants and flies. Many animals in nature use Levy flight strategy as an ideal way of foraging. By studying this phenomenon, French mathematician Paul Pierre Levy proposed the mathematical model of Levy flight [43]. Levy flight is an operator conforming to Levy distribution. The step size of Levy flight is random and mixed with long and short distances, which makes it easier to search over a large scale and with unknown scope compared to Brownian motion [44]. In the searching process, the Levy operator often uses short steps to walk and occasionally uses long steps to jump, which allows it to efficiently get rid of the effects of local attraction points. Therefore, in the random searching problem, many heuristic algorithms adopt this strategy to modify the iterative process, which efficiently helps the algorithm to get rid of the influence of local attraction points [45–47].

The Levy distribution can be expressed by the following mathematical model:

$$L(s) \sim |s|^{-1-\beta} \tag{11}$$

where $\beta$ is in the range of (0, 2), $s$ is the step size, and $L(s)$ is the probability density of a step size, $s$, according to Levy modeling. The mathematical modeling of Levy distribution is as follows:

$$L(s,\gamma,\mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}}\exp[-\frac{\gamma}{2(s-\mu)}]\frac{1}{(s-\mu)^{3/2}}, 0<\mu<s<\infty \\ 0 \qquad\qquad\qquad\qquad\qquad\quad , otherwise \end{cases} \tag{12}$$

where $\mu$ represents the minimum step size and $\mu > 0$, $\gamma$ represents size parameters. When $s \to \infty$, Equation (12) can be written in the following form:

$$L(s,\gamma,\mu) \approx \sqrt{\frac{\gamma}{2\pi}}\frac{1}{s^{3/2}} \tag{13}$$

Usually, scholars regard $L(s)$ approximation as the following mathematical formula:

$$L(s) \to \frac{\alpha\beta\cdot\Gamma(\beta)\sin(\pi\beta/2)}{\pi|s|^{1+\beta}}, s \to \infty \tag{14}$$

where $\Gamma$ represents gamma function. Its mathematical model is as follows:

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt \tag{15}$$

Due to the high complexity of Levy distribution, researchers often use the Mantegna [48] algorithm to simulate Levy flight step size, $s$, which is defined as follows:

$$s = \frac{\mu}{|\nu|^{1/\beta}} \tag{16}$$

where $\mu$ and $v$ are defined as follows:

$$\mu \sim N\left(0, \sigma_\mu^2\right) \tag{17}$$

$$\nu \sim N\left(0, \sigma_\nu^2\right) \tag{18}$$

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left[\frac{(1+\beta)}{2}\right] \cdot \beta \cdot 2^{\frac{(1+\beta)}{2}}} \right\}, \ \sigma_\nu = 1 \tag{19}$$

where the value of $\beta$ is usually 1.5.

To show the global exploration capability of Levy flight more intuitively, this paper compares Levy flight with random walk strategy. The simulation steps of Levy flight and random walk are set to 300. The comparison results are presented in Figure 3.

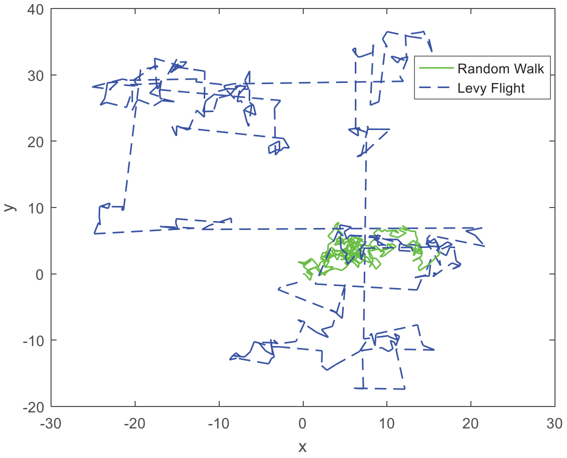


**Figure 3.** Simulation comparison experiment diagram of Levy flight and random walk.

Figure 3 shows that the Levy flight has a larger search range than random walk. The jump points of the random walk strategy are more concentrated, and the jump points of the Levy flight strategy are widely distributed. Figure 3 fully demonstrates the characteristics of Levy flight, which can make it better to explore in the whole searching space.

### 3.3. Nonlinear Adaptive Weight

How to balance the exploration capability and the exploration capability of the swarm intelligence optimization algorithm is very important. Weight parameters play an important role in the TSO algorithm. When the tuna chooses the spiral foraging strategy, in Equations (5) and (6), the weight parameters $\alpha_1$ and $\alpha_2$ determine the degree of how much tuna individuals follow the optimal individual to forage. This reflects the optimization process of the algorithm. Similarly, in the parabolic foraging strategy, the weight parameter $p$ in Equation (2) determines the degree of how much ordinary individuals follow the optimal individual. When the weight parameter is large, the degree of tuna following the optimal individual is higher, which makes the whole tuna population better explore the whole space. When the weight parameter is small, ordinary tuna individuals do not follow the optimal individuals. They will swim around a small part of the space, which facilitates the ordinary tuna individual to develop the field around itself. To sum up, the exploration and the exploitation capabilities of TSO depend on the changes of weight parameters $\alpha_1$, $\alpha_2$, and $p$.

From Equations (5) and (6), it can be seen that the weight parameters $\alpha_1$ and $\alpha_2$ are linear changes. However, the optimization process of TSO is very complex, and the linear changes of weight parameters $\alpha_1$ and $\alpha_2$ cannot reflect the actual optimization process of the algorithm. Nowadays, in order to overcome the drawbacks caused by linear control weights, many scholars use nonlinear adaptive weights to improve the swarm intelligence optimization algorithms [49–51]. Repeated experiments indicate that the optimization effect of the nonlinear adaptive weight strategy is better than the linear weight strategy. Therefore, two improved nonlinear weight parameters $\alpha_{1i}$ and $\alpha_{2i}$ are introduced in this paper. Their mathematical models are as follows:

$$\alpha_{1i}(t) = \alpha_{1ini} - (\alpha_{1ini} - \alpha_{1fin}) \cdot \sin(\frac{t}{\mu \cdot T_{Max}} \cdot \pi) \tag{20}$$

$$\alpha_{2i}(t) = \alpha_{2ini} - (\alpha_{2ini} - \alpha_{2fin}) \cdot \sin(\frac{t}{\mu \cdot T_{Max}} \cdot \pi) \tag{21}$$

where $\mu$ = 2, $\alpha_{1ini}$ denotes the initial value of $\alpha_1$, $\alpha_{1fin}$ denotes the final value of $\alpha_1$, $\alpha_{2ini}$ denotes the initial value of $\alpha_2$, and $\alpha_{2fin}$ denotes the final value of $\alpha_2$. We compared the improved weight parameters $\alpha_{1i}$ and $\alpha_{2i}$ with the original weight parameters $\alpha_1$ and $\alpha_2$. The results are displayed in Figure 4. In the experiment, $T_{\text{Max}}$ = 500.

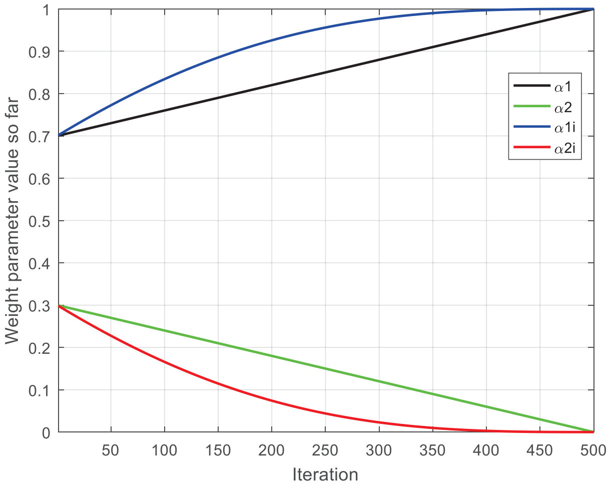


**Figure 4.** Comparison of weight coefficients $\alpha_1$ and $\alpha_2$ before and after improvement.

It can be clearly seen from Figure 4 that the improved weight parameters $\alpha_{1i}$ and $\alpha_{2i}$ change rapidly in the early stage, which makes ordinary tuna individuals more closely follow the optimal individual. It increases the global exploration capability of TSO. The weight parameters $\alpha_{1i}$ and $\alpha_{2i}$ change slowly in the late stage, which enables tuna individuals to explore their surrounding areas. It increases the local search capability of TSO.

In the spiral foraging strategy, a new nonlinear weight parameter $p_i$ is proposed. Its mathematical model is as follows:

$$p_i(t) = p_{ini} - (p_{ini} - p_{fin}) \cdot \sin(\frac{t}{\mu \cdot T_{Max}} \cdot \pi) \tag{22}$$

where $p_{ini}$ represents the initial value of $p$, and $p_{fin}$ represents the final value of $p$. We compare the improved weight parameter ${p_i}^2$ with the original weight parameter $p^2$. The comparison curves are displayed in Figure 5. In the comparison curve, $T_{\text{Max}}$ = 500.

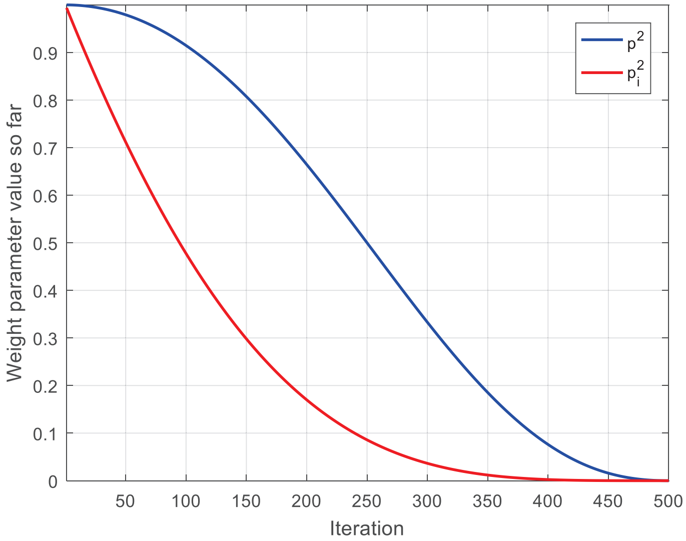


**Figure 5.** Comparison of the weight coefficient $p$ before and after the improvement.

As can be seen from Figure 5, the improved weight parameter $p_i{}^2$ decreases rapidly in the early stage, so a tuna individual can follow its previous individual more closely. It increases the global exploration capability of TSO. The improved weight parameter $p_i{}^2$ decreases slowly in the late iteration, so tuna individuals can swim and explore in the surrounding space. It increases the local exploration capability of TSO.

### *3.4. Improved Nonlinear Tuna Swarm Optimization Algorithm Based on Circular Chaotic Map and Levy Flight Operator*

The TSO algorithm usually uses random data to initialize population in solving function optimization problems, which may lead to the phenomenon that candidate solutions are clustered together. However, this phenomenon will lead to poor population diversity, which eventually leads to poor optimization results of the algorithm. Circle chaotic map has the advantages of randomness and ergodicity. In the optimization process of TSO, these advantages make it easier for the algorithm to escape the attraction of local extremum, and helps the algorithm to maintain the diversity of the swarm. Therefore, an improved Circle chaotic map strategy is introduced to initialize the tuna swarm. The swarm initialization mechanism is upgraded from Equations (1)–(10).

For the swarm intelligence optimization algorithm, how to get rid of the influence of local attraction points is a very important issue. The Levy flight strategy is an operator that can strengthen the global capability of TSO. This mechanism often uses short steps to walk and occasionally uses long steps to jump. The low-frequency use of long step length can ensure that TSO can extensively search the entire search area. The high-frequency use of short step length can ensure that TSO can locally search its nearest scope. Therefore, this paper introduces the Levy operator to modify the swarm update strategy of TSO. Considering that the jump of the Levy operator is too intense, and it may jump out of the main range in the process of operation, this paper adds step control parameters on the basis of the original Levy operator. The small step size control parameters can control the search of TSO in a small scope, which can enhance the local exploration ability of TSO without weakening the global exploration ability. The step size control parameters with large values can control the exploration of TSO in a large scope, which is conducive to solving the complex optimization problem.

The original TSO designed the parabolic foraging strategy and the spiral foraging strategy to balance the global exploration and the local exploitation capabilities of TSO. However, in the spiral foraging strategy, the linear changes of the weight parameters $\alpha_1$ and $\alpha_2$ cannot solve the actual complex problems well. In the parabolic foraging strategy, the change of the weight parameter $p$ cannot effectively provide the solution to TSO for the global and the local exploration abilities. This paper uses nonlinear adaptive weight to modify the spiral foraging strategy and parabolic foraging strategy in TSO. The mathematical model of weight parameter $p_i$ is upgraded from Equation (3) to Equation (22), and the mathematical models of $\alpha_{1i}$ and $\alpha_{2i}$ are upgraded from Equations (5) and (6) to Equations (20) and (21), respectively.

The mathematical model of the improved spiral foraging strategy based on the Levy operator and nonlinear adaptive weight strategy is as follows:

$$X_i^{t+1} = \begin{cases} \alpha_{1i} \cdot (X_{rand}^t + L\tau \cdot \left| X_{rand}^t - X_i^t \right| + \alpha_{2i} \cdot X_i^t), \\ \qquad i = 1 & ,if \text{ rand} < \frac{t}{t_{\max}} \\ \alpha_{1i} \cdot (X_{rand}^t + L\tau \cdot \left| X_{rand}^t - X_i^t \right| + \alpha_{2i} \cdot X_{i-1}^t), \\ \qquad i = 2, 3, \ldots, NP \\ \alpha_{1i} \cdot (X_{best}^t + L\tau \cdot \left| X_{best}^t - X_i^t \right| + \alpha_{2i} \cdot X_i^t), \\ \qquad i = 1 & ,if \text{ rand} < \frac{t}{t_{\max}} \\ \alpha_{1i} \cdot (X_{best}^t + L\tau \cdot \left| X_{best}^t - X_i^t \right| + \alpha_{2i} \cdot X_{i-1}^t), \\ \qquad i = 2, 3, \ldots, NP \end{cases} \tag{23}$$

where $L\tau$ is an improved distance control parameter combined with the Levy operator. Its mathematical model is as follows:

$$L\tau = e^{\alpha \cdot Levy(s) \cdot l} \cdot \cos(2\pi \cdot Levy(s) \cdot \alpha) \tag{24}$$

where $Levy(s)$ is the step size of the Lévy operator, and $\alpha$ is the step size control coefficient. In this article, $\alpha$ = 0.01. The mathematical model of improved parabolic foraging strategy based on the Levy operator and nonlinear adaptive weight strategy is as follows:

$$X_i^{t+1} = \begin{cases} X_{best}^t + \alpha \cdot Levy(s) \cdot (X_{best}^t - X_i^t) + TF \cdot p^2 \cdot (X_{best}^t - X_i^t), & \text{if rand } < \ 0.5 \\ TF \cdot {p_i}^2 \cdot X_i^t, & \text{if rand} \geq 0.5 \end{cases} \tag{25}$$

Based on the above improvement strategies, an improved TSO is proposed, called CLTSO. The pseudocode of CLTSO is shown in Algorithm 2, and the process diagram of CLTSO is shown in Figure 6.

**Algorithm 2** Pseudocode of CLTSO Algorithm

```
Initialization: Set parameters $NP$, $Dim$, $a$, $z$ and $T_{\text{Max.}}$ Initialize the position of tuna $X_i$ ($i$ = 1, 2, . . . , $NP$) by (10)
Counter $t$ = 0
while $T$ < $T_{\text{Max}}$ do
   Calculate the fitness value of all tuna
   Update the position and value of the best tuna $X_{best}^t$
   for (each tuna) do
          Update $\alpha_{1i}$, $\alpha_{2i}$, $p_i$ by (20), (21), (22)
             if ($rand$ < $z$) then
                  Update $X_i^{t+1}$ by (10)
           else if ($rand \geq z$) then
                  if ($rand$ < 0.5) then
                       Update $X_i^{t+1}$ by (23)
                 else if ($rand \geq$ 0.5) then
                       Update $X_i^{t+1}$ by (25)
$t$ = $t$ + 1
return the best fitness value f ($X_{best}$) and the best tuna $X_{best}$
```

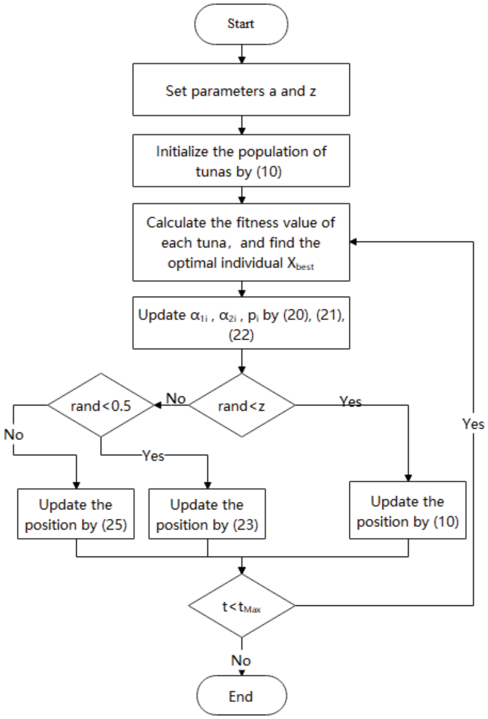

Start
Set parameters a and z
Initialize the population of tunas by (10)
Calculate the fitness value of each tuna, and find the optimal individual $X_{best}$
Update $\alpha_{1i}$, $\alpha_{2i}$, $p_i$ by (20), (21), (22)
rand<0.5
No
rand<z
Yes
Yes
No
Yes
Update the position by (25)
Update the position by (23)
Update the position by (10)
$t<t_{Max}$
No
End

**Figure 6.** Flow chart of CLTSO.

Comparing Algorithms 1 and 2, it is clear that the overall structures are similar. The update strategies have been changed. Therefore, the improved operator proposed in this paper does not destroy the structural simplicity of the original TSO algorithm.

### *3.5. Time Complexity Analysis*

Time complexity is an important measurement tool for evaluating the efficiency of an algorithm. In much of the research literature, it is represented by the symbol $O$. The time complexity is closely related to the number of instruction operations of the algorithm. The time complexity of TSO is closely related to iteration times, location update mechanism, and the evaluation times of fitness value function. The time complexity of CLTSO is closely related to the number of iterations, the number of fitness function evaluations, and the improvement operator. To compare the time cost differences between TSO and CLTSO, the time complexity of TSO and CLTSO is evaluated as follows. The time complexity of each operation instruction in the TSO is discussed below.

1. Initialize $N$ individuals in the TSO, each with a dimension of $D$, so $N \cdot D$ calculations are required.
2. Calculate the fitness value of each individual in the tuna population and select the optimal individual in the current population. Therefore, it needs to calculate $[N \cdot (N-1)]/2$ times.
3. Update the values of parameters $\alpha_1$, $\alpha_2$, and $p$, which are computed 3 times.
4. Update all tuna individuals in the search space, which are computed $N \cdot D$ times.
5. Return the best individual, $X_{best}$, in the tuna population, which requires this code to be executed 1 time.

The instructions in steps 2 to 4 need to be iteratively run $T_{\text{Max}}$ times. Combining the above analysis process, the time complexity of TSO can be expressed as $O(TSO)$ = $T_{Max} \cdot [(N^2 - N)/2 + N \cdot D + 3]$.

The time complexity of each operation instruction in CLTSO is analyzed as follows.

1. Initialize $N$ individuals in the CLTSO, each with a dimension of $D$, so $N \cdot D$ calculations are required.
2. Calculate the fitness value of each individual in the tuna population and select the optimal individual in the current population. Therefore, it needs to be calculated $[N \cdot (N-1)]/2$ times.
3. Update the values of parameters $\alpha_{1i}$, $\alpha_{2i}$, and $p_i$, which needs to be calculated 3 times.
4. Update all tuna individuals in the search space, which needs to be calculated $N \cdot D$ times.
5. When each individual in the tuna population is updated, the Levy operator needs to be calculated 1 time. Therefore, it needs to be run $N$ times in total.
6. Return the best individual, $X_{best}$, in the tuna population, which requires this code to be executed 1 time.

Steps 2 to 5 require a total of $T_{\text{Max}}$ iterations. Therefore, the time complexity of CLTSO can be expressed as $O(CLTSO) = T_{Max} \cdot [(N^2 - N)/2 + N \cdot D + 3 + N]$.

Compared with the tuna swarm optimization algorithm, the three operators proposed in this paper slightly increase the time cost. CLTSO and TSO have very close time complexity.

## 4. Simulation Experiments and Results Analysis

To verify the effectiveness of the proposed CLTSO in solving different optimization problems, in this section, 22 benchmark functions are applied to design a series of experiments to compare CLTSO with other famous meta-heuristic algorithms. In addition, to illustrate the outstanding performance of CLTSO, we compared it against the tuna swarm optimization algorithm (TSO), the improved TSO based on the Levy flight operator (LTSO), the improved TSO based on the Circle chaotic map, and nonlinear adaptive weights (CTSO). Finally, this section provides a detailed analysis of the experimental results.

### *4.1. Benchmark Function*

Twenty-two different types of benchmark functions are selected to evaluate the capability of CLTSO, which cover unimodal, multimodal, fixed-dimension multimodal, and combined functions in the CEC2014 [52]. Through a survey of relevant literature, we find that CEC2014 is a classic test function, so it can be used as a benchmark to evaluate the performance of the proposed algorithm. Its mathematical model is given in Table 2. $F_1$~$F_7$ are unimodal functions, which are used to evaluate the convergence rate of the algorithm. $F_8$~$F_{14}$ are multimodal functions, which are applied to verify whether the algorithm has good global exploration capability. $F_{15}$~$F_{22}$ are the CEC2014 functions, which are applied to test the comprehensive capability of these algorithms.

### *4.2. Comparison Algorithm and Parameter Setting*

Based on these 22 benchmark functions, a series of comparative experiments are designed to test the selected algorithms, which include Accelerated Particle Swarm Optimization (APSO) [53], WOA, the Fitness-Distance Balance based adaptive guided differential evolution (FDB-AGDE) algorithm [54], Covariance Matrix Adaptation Evolutionary Strategies (CMA-ES) [55], TSO, and CLTSO. The parameter values of the algorithms involved in these experiments are shown in Table 3. The symbol ' ~ ' indicates that the algorithm does not set parameter values. Functions $F_1$~$F_{13}$ are tested in 30 and 100 dimensions, respectively, and $F_{14}$ is tested in its suitable dimension. Eight CEC2014 benchmark functions are tested in 50 dimensions. The maximum number of evaluations of $F_1$~$F_{14}$ are 1000. Because CEC benchmark functions are complex, the number of evaluations of 8 CEC2014 functions are simplified to 5000 without losing representativeness. The swarm size of each algorithm is

30. To avoid accidental interference, we run each algorithm 30 times independently in each experiment.

**Table 2.** Benchmark functions.

| Function | Dim | Range | $\mathbf{f_{min}}$ |
|---|---|---|---|
| $F_1(x) = \sum_{i=1}^{D} x_i^2$ | 30,100 | $[-100, 100]$ | 0 |
| $F_2(x) = \sum\limits_{i=1}^{D} \lvert x_i \rvert + \prod\limits_{i=1}^{D} \lvert x_i \rvert$ | 30,100 | $[-10, 10]$ | 0 |
| $F_3(x) = \sum\limits_{j=1}^{D} \left( \sum\limits_{i=1}^{j} x_i \right)^2$ | 30,100 | $[-100, 100]$ | 0 |
| $F_4(x) = \max_i \{ \lvert x_i \rvert, 1 \le i \le D \}$ | 30,100 | $[-100, 100]$ | 0 |
| $F_5(x) = \sum_{i=1}^{D} 100 (x_{i+1}^2 - x_i^2)^2 + (x_i - 1)^2$ | 30,100 | $[-30, 30]$ | 0 |
| $F_6(x) = \sum\limits_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$ | 30,100 | $[-100, 100]$ | 0 |
| $F_7(x) = \sum\limits_{i=1}^{D} i x_i{}^4 + \text{random}[0, 1)$ | 30,100 | $[-1.28, 1.28]$ | 0 |
| $F_8(x) = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$ | 30,100 | $[-500, 500]$ | $-418.$<br>$9829 \times \mathrm{D}$ |
| $F_9(x) = \sum_{i=1}^{D} (x_i^2 - 10 \cos(2\pi x_i) + 10)$ | 30,100 | $[-5.12, 5.12]$ | 0 |
| $F_{10}(x) = -20 \exp\left( -0.2 \sqrt{(1/D) \sum_{i=1}^{D} x_i^2} \right) - \exp\left( (1/D) \sum_{i=1}^{D} \cos(2\pi x_i) \right) + 20 + \exp(1)$ | 30,100 | $[-32, 32]$ | $8.8818 \times 10^{-16}$ |
| $F_{11}(x) = (1/4000) \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left( x_i / \sqrt{i} \right) + 1$ | 30,100 | $[-600, 600]$ | 0 |
| $F_{12}(x) = \pi / D \left\{ 10 \sin^2(\pi y_i) + \sum_{i=1}^{D} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1) \right\}$<br>$+ \sum_{i=1}^{D} u(x_i, 10, 100, 4)$ | 30,100 | $[-50, 50]$ | 0 |
| $\mathrm{y}_i = 1 + [(x_i + 1)/4] u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a < x_i < a \\ k(-x_i - a)^m, x_i < a \end{cases}$ | | | |
| $F_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_i) + \sum_{i=1}^{D} (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\}$<br>$+ \sum_{i=1}^{D} u(x_i, 5, 100, 4) \Big\}$ | 30,100 | $[-50, 50]$ | 0 |
| $F_{14}(x) = \left( (1/500) + \sum_{j=1}^{25} \left( 1 / (j + \sum_{i=1}^{2} (x_i - a_{ij})^6) \right) \right)^{-1}$ | 2 | $[-65.53, 65.53]$ | 0.998004 |
| $F_{15}(x)(CEC2014\ 1: \text{Rotated High Conditioned Elliptic Function})$ | 50 | $[-100, 100]$ | 100 |
| $F_{16}(x)(CEC2014\ 2: \text{Rotated Bent Cigar Function})$ | 50 | $[-100, 100]$ | 200 |
| $F_{17}(x)(CEC2014\ 3: \text{Rotated Discus Function})$ | 50 | $[-100, 100]$ | 300 |
| $F_{18}(x)(CEC2014\ 5: \text{Shifted and Rotated Rosenbrock})$ | 50 | $[-100, 100]$ | 500 |
| $F_{19}(x)(CEC2014\ 18: \text{Shifted and Rotated Expanded Scaffer's F6 Function})$ | 50 | $[-100, 100]$ | 1800 |
| $F_{20}(x)(CEC2014\ 20: \text{Hybrid Function 4 (N = 4)})$ | 50 | $[-100, 100]$ | 2000 |
| $F_{21}(x)(CEC2014\ 21: \text{Hybrid Function 5 (N = 5)})$ | 50 | $[-100, 100]$ | 2100 |
| $F_{22}(x)(CEC2014\ 30: \text{Composition Function 8 (N = 3)})$ | 50 | $[-100, 100]$ | 3000 |

**Table 3.** Parameter values of the algorithms.

| Algorithm | Parameter Value |
|---|---|
| APSO | $\alpha = 1,\ \beta = 0.5,\ \gamma = 0.95$ |
| WOA | $l \in (-1, 1)$ |
| FDB-AGDE | $\mu_{CR} = 0.5$ |
| CMA-ES | $\mu = 2$ |
| TSO | $a = 0.7,\ z = 0.05$ |
| CLTSO | $a = 0.7,\ z = 0.05$ |
| CTSO | $a = 0.7,\ z = 0.05$ |
| LTSO | $a = 0.7,\ z = 0.05$ |

### 4.3. Results and Analysis

Table 4 shows the experimental results of CLTSO and other algorithms in low dimensional benchmark functions (dimension = 30), where Std is standard deviation and Mean is mean value. Mean represents the solution accuracy of these algorithms. Std reflects the stability of these algorithms in the solution process. $F_{14}$ is tested in its own dimension. Table 5 displays the experimental results of CLTSO and other algorithms in high dimensional benchmark functions (dimension = 100). The experimental results of eight composite functions in CEC2014 are displayed in Table 6.

**Table 4.** Experimental results in 30 dimensions.

| Function | Performance | APSO | WOA | FDB-AGDE | CMA-ES | TSO | CLTSO |
|---|---|---|---|---|---|---|---|
| $F_1$ | Mean | $5.09 \times 10^{-39}$ | $4.82 \times 10^{-150}$ | $1.23 \times 10^{-8}$ | $5.99 \times 10^{-15}$ | 0 | 0 |
| | Std | $1.05 \times 10^{-39}$ | $2.59 \times 10^{-149}$ | $1.34 \times 10^{1}$ | $3.94 \times 10^{-15}$ | 0 | 0 |
| $F_2$ | Mean | $4.38 \times 10^{-1}$ | $8.02 \times 10^{-103}$ | $4.49 \times 10^{-6}$ | $1.83 \times 10^{-7}$ | $1.71 \times 10^{-252}$ | 0 |
| | Std | $5.79 \times 10^{-1}$ | $3.63 \times 10^{-102}$ | $5.61 \times 10^{1}$ | $4.53 \times 10^{-8}$ | 0 | 0 |
| $F_3$ | Mean | $1.32 \times 10^{1}$ | $2.01 \times 10^{4}$ | $1.08 \times 10^{-96}$ | $6.14 \times 10^{-6}$ | 0 | 0 |
| | Std | $4.84 \times 10^{0}$ | $9.41 \times 10^{3}$ | $6.35 \times 10^{4}$ | $1.16 \times 10^{-5}$ | 0 | 0 |
| $F_4$ | Mean | $4.96 \times 10^{-1}$ | $3.61 \times 10^{1}$ | $1.64 \times 10^{0}$ | $8.37 \times 10^{-6}$ | $6.42 \times 10^{-249}$ | 0 |
| | Std | $1.75 \times 10^{-1}$ | $2.69 \times 10^{1}$ | $1.45 \times 10^{1}$ | $2.29 \times 10^{-6}$ | 0 | 0 |
| $F_5$ | Mean | $5.02 \times 10^{1}$ | $2.72 \times 10^{1}$ | $1.50 \times 10^{2}$ | $6.64 \times 10^{1}$ | $2.94 \times 10^{-4}$ | $2.12 \times 10^{-4}$ |
| | Std | $4.56 \times 10^{1}$ | $4.96 \times 10^{-1}$ | $1.14 \times 10^{2}$ | $1.52 \times 10^{2}$ | $7.65 \times 10^{-1}$ | $3.82 \times 10^{-5}$ |
| $F_6$ | Mean | $4.01 \times 10^{-32}$ | $8.72 \times 10^{-2}$ | $1.09 \times 10^{-8}$ | $6.59 \times 10^{-15}$ | $1.37 \times 10^{-9}$ | $2.04 \times 10^{-10}$ |
| | Std | $1.60 \times 10^{-32}$ | $9.95 \times 10^{-2}$ | $1.33 \times 10^{1}$ | $3.44 \times 10^{-15}$ | $8.89 \times 10^{-6}$ | $2.40 \times 10^{-10}$ |
| $F_7$ | Mean | $1.54 \times 10^{-1}$ | $1.53 \times 10^{-3}$ | $2.36 \times 10^{-2}$ | $2.44 \times 10^{-2}$ | $2.16 \times 10^{-5}$ | $1.81 \times 10^{-5}$ |
| | Std | $2.03 \times 10^{-2}$ | $2.05 \times 10^{-3}$ | $9.40 \times 10^{0}$ | $6.71 \times 10^{-3}$ | $2.19 \times 10^{-4}$ | $6.32 \times 10^{-5}$ |
| $F_8$ | Mean | $-1.09 \times 10^{2}$ | $-1.16 \times 10^{4}$ | $-1.26 \times 10^{4}$ | $-4.41 \times 10^{11}$ | $-8.38 \times 10^{2}$ | $-1.26 \times 10^{4}$ |
| | Std | $3.25 \times 10^{0}$ | $1.50 \times 10^{3}$ | $3.32 \times 10^{-1}$ | $2.34 \times 10^{12}$ | $1.17 \times 10^{4}$ | $6.00 \times 10^{-8}$ |
| $F_9$ | Mean | $7.42 \times 10^{1}$ | 0 | $3.11 \times 10^{1}$ | $5.60 \times 10^{1}$ | 0 | 0 |
| | Std | $5.96 \times 10^{0}$ | 0 | $1.62 \times 10^{1}$ | $6.32 \times 10^{1}$ | 0 | 0 |
| $F_{10}$ | Mean | $5.36 \times 10^{-1}$ | $4.56 \times 10^{-15}$ | $6.78 \times 10^{-7}$ | $7.01 \times 10^{-1}$ | $8.88 \times 10^{-16}$ | $8.88 \times 10^{-16}$ |
| | Std | $5.28 \times 10^{-1}$ | $2.15 \times 10^{-15}$ | $3.73 \times 10^{-1}$ | $3.78 \times 10^{0}$ | $8.29 \times 10^{-16}$ | 0 |
| $F_{11}$ | Mean | $8.49 \times 10^{-3}$ | $1.61 \times 10^{-3}$ | $2.85 \times 10^{-7}$ | $3.29 \times 10^{-4}$ | 0 | 0 |
| | Std | $1.67 \times 10^{-2}$ | $8.69 \times 10^{-3}$ | $1.64 \times 10^{1}$ | $1.77 \times 10^{-3}$ | 0 | 0 |
| $F_{12}$ | Mean | $1.08 \times 10^{-1}$ | $6.17 \times 10^{-3}$ | $1.74 \times 10^{-25}$ | $2.01 \times 10^{-15}$ | $2.65 \times 10^{-10}$ | $6.75 \times 10^{-14}$ |
| | Std | $1.21 \times 10^{-1}$ | $6.67 \times 10^{-3}$ | $2.01 \times 10^{1}$ | $1.14 \times 10^{-15}$ | $1.14 \times 10^{-7}$ | $3.87 \times 10^{-11}$ |
| $F_{13}$ | Mean | $2.38 \times 10^{-3}$ | $3.11 \times 10^{-1}$ | $1.87 \times 10^{-19}$ | $3.77 \times 10^{-14}$ | $5.12 \times 10^{-8}$ | $1.24 \times 10^{-9}$ |
| | Std | $4.42 \times 10^{-3}$ | $2.72 \times 10^{-1}$ | $1.54 \times 10^{1}$ | $3.15 \times 10^{-14}$ | $2.84 \times 10^{-3}$ | $3.77 \times 10^{-9}$ |
| $F_{14}$ | Mean | $1.27 \times 10^{1}$ | $2.27 \times 10^{0}$ | $9.98 \times 10^{-1}$ | $7.65 \times 10^{0}$ | $9.98 \times 10^{-1}$ | $9.98 \times 10^{-1}$ |
| | Std | $1.12 \times 10^{-13}$ | $2.91 \times 10^{0}$ | $2.71 \times 10^{0}$ | $3.59 \times 10^{0}$ | $9.31 \times 10^{-1}$ | $2.69 \times 10^{-16}$ |

**Table 5.** Experimental results in 100 dimensions.

| Function | Performance | APSO | WOA | FDB-AGDE | CMA-ES | TSO | CLTSO |
|---|---|---|---|---|---|---|---|
| $F_1$ | Mean | $1.84 \times 10^{1}$ | $1.03 \times 10^{-149}$ | $7.27 \times 10^{1}$ | $1.83 \times 10^{-3}$ | 0 | 0 |
| | Std | $1.40 \times 10^{0}$ | $4.01 \times 10^{-149}$ | $1.98 \times 10^{1}$ | $4.15 \times 10^{-4}$ | 0 | 0 |
| $F_2$ | Mean | $4.13 \times 10^{1}$ | $6.59 \times 10^{-102}$ | $7.95 \times 10^{0}$ | $2.99 \times 10^{-1}$ | $8.66 \times 10^{-235}$ | 0 |
| | Std | $2.74 \times 10^{0}$ | $2.42 \times 10^{-101}$ | $7.89 \times 10^{0}$ | $1.09 \times 10^{-1}$ | 0 | 0 |
| $F_3$ | Mean | $2.23 \times 10^{2}$ | $8.92 \times 10^{5}$ | $7.05 \times 10^{-86}$ | $4.80 \times 10^{5}$ | 0 | 0 |
| | Std | $1.40 \times 10^{1}$ | $2.09 \times 10^{5}$ | $1.05 \times 10^{1}$ | $1.31 \times 10^{5}$ | 0 | 0 |
| $F_4$ | Mean | $2.29 \times 10^{0}$ | $7.06 \times 10^{1}$ | $5.91 \times 10^{1}$ | $1.73 \times 10^{0}$ | $5.52 \times 10^{-229}$ | 0 |
| | Std | $8.35 \times 10^{-2}$ | $2.77 \times 10^{1}$ | $6.11 \times 10^{0}$ | $3.03 \times 10^{-1}$ | 0 | 0 |
| $F_5$ | Mean | $6.22 \times 10^{3}$ | $9.77 \times 10^{1}$ | $1.30 \times 10^{5}$ | $3.59 \times 10^{2}$ | $1.08 \times 10^{-1}$ | $1.89 \times 10^{-3}$ |
| | Std | $2.28 \times 10^{3}$ | $4.05 \times 10^{-1}$ | $9.66 \times 10^{5}$ | $1.49 \times 10^{3}$ | $1.99 \times 10^{-1}$ | $4.41 \times 10^{-3}$ |
| $F_6$ | Mean | $2.66 \times 10^{1}$ | $1.76 \times 10^{0}$ | $5.27 \times 10^{-5}$ | $1.71 \times 10^{-3}$ | $5.10 \times 10^{-5}$ | $4.65 \times 10^{-5}$ |
| | Std | $7.14 \times 10^{0}$ | $6.30 \times 10^{-1}$ | $1.37 \times 10^{1}$ | $3.09 \times 10^{-4}$ | $2.37 \times 10^{-2}$ | $7.68 \times 10^{-5}$ |
| $F_7$ | Mean | $1.83 \times 10^{3}$ | $1.67 \times 10^{-3}$ | $2.81 \times 10^{-1}$ | $1.39 \times 10^{-1}$ | $2.76 \times 10^{-4}$ | $1.04 \times 10^{-4}$ |
| | Std | $6.20 \times 10^{2}$ | $1.19 \times 10^{-3}$ | $1.37 \times 10^{1}$ | $1.83 \times 10^{-2}$ | $3.08 \times 10^{-4}$ | $1.13 \times 10^{-4}$ |
| $F_8$ | Mean | $-2.35 \times 10^{2}$ | $-3.73 \times 10^{4}$ | $-3.45 \times 10^{4}$ | $-1.81 \times 10^{5}$ | $-2.79 \times 10^{3}$ | $-4.19 \times 10^{4}$ |
| | Std | $9.35 \times 10^{0}$ | $5.59 \times 10^{3}$ | $4.00 \times 10^{3}$ | $3.12 \times 10^{4}$ | $3.91 \times 10^{4}$ | $2.95 \times 10^{-3}$ |
| $F_9$ | Mean | $4.33 \times 10^{2}$ | 0 | $2.25 \times 10^{2}$ | $6.69 \times 10^{2}$ | 0 | 0 |
| | Std | $2.60 \times 10^{1}$ | 0 | $1.18 \times 10^{2}$ | $1.64 \times 10^{2}$ | 0 | 0 |
| $F_{10}$ | Mean | $3.58 \times 10^{0}$ | $4.20 \times 10^{-15}$ | $5.86 \times 10^{0}$ | $9.41 \times 10^{-3}$ | $8.88 \times 10^{-16}$ | $8.88 \times 10^{-16}$ |
| | Std | $3.04 \times 10^{-1}$ | $2.23 \times 10^{-15}$ | $4.11 \times 10^{0}$ | $1.04 \times 10^{1}$ | $8.29 \times 10^{-16}$ | 0 |
| $F_{11}$ | Mean | $4.39 \times 10^{-1}$ | 0 | $1.71 \times 10^{0}$ | $3.49 \times 10^{-2}$ | 0 | 0 |
| | Std | $6.34 \times 10^{-2}$ | 0 | $1.12 \times 10^{1}$ | $6.78 \times 10^{-3}$ | 0 | 0 |
| $F_{12}$ | Mean | $5.46 \times 10^{-1}$ | $1.80 \times 10^{-2}$ | $6.02 \times 10^{-3}$ | $2.12 \times 10^{-4}$ | $2.49 \times 10^{-8}$ | $2.78 \times 10^{-9}$ |
| | Std | $1.09 \times 10^{-1}$ | $7.22 \times 10^{-3}$ | $9.60 \times 10^{0}$ | $5.70 \times 10^{-5}$ | $5.86 \times 10^{-5}$ | $2.19 \times 10^{-7}$ |
| $F_{13}$ | Mean | $8.73 \times 10^{0}$ | $1.65 \times 10^{0}$ | $5.93 \times 10^{-1}$ | $3.76 \times 10^{-3}$ | $2.02 \times 10^{-4}$ | $6.80 \times 10^{-6}$ |
| | Std | $2.13 \times 10^{0}$ | $7.21 \times 10^{-1}$ | $1.53 \times 10^{1}$ | $2.83 \times 10^{-3}$ | $4.38 \times 10^{-3}$ | $7.14 \times 10^{-6}$ |

**Table 6.** Simulation results of CEC2014 functions.

| Function | Performance | APSO | WOA | FDB-AGDE | CMA-ES | TSO | CLTSO |
|---|---|---|---|---|---|---|---|
| $F_{15}$ | Mean | $1.19 \times 10^{10}$ | $8.85 \times 10^{8}$ | $3.36 \times 10^{5}$ | $1.84 \times 10^{7}$ | $2.22 \times 10^{6}$ | $4.35 \times 10^{5}$ |
| | Std | $3.12 \times 10^{7}$ | $3.34 \times 10^{8}$ | $7.01 \times 10^{7}$ | $3.94 \times 10^{6}$ | $1.21 \times 10^{6}$ | $4.35 \times 10^{5}$ |
| $F_{16}$ | Mean | $1.65 \times 10^{11}$ | $7.71 \times 10^{10}$ | $3.36 \times 10^{2}$ | $2.02 \times 10^{4}$ | $1.00 \times 10^{4}$ | $2.75 \times 10^{2}$ |
| | Std | $3.25 \times 10^{8}$ | $7.86 \times 10^{9}$ | $1.20 \times 10^{1}$ | $5.08 \times 10^{4}$ | $4.94 \times 10^{9}$ | $1.15 \times 10^{4}$ |
| $F_{17}$ | Mean | $1.99 \times 10^{8}$ | $9.77 \times 10^{4}$ | $7.36 \times 10^{2}$ | $8.33 \times 10^{5}$ | $8.38 \times 10^{3}$ | $3.59 \times 10^{2}$ |
| | Std | $3.10 \times 10^{3}$ | $9.92 \times 10^{3}$ | $1.09 \times 10^{1}$ | $1.18 \times 10^{5}$ | $7.60 \times 10^{3}$ | $3.29 \times 10^{1}$ |
| $F_{18}$ | Mean | $5.20 \times 10^{2}$ | $5.21 \times 10^{2}$ | $5.21 \times 10^{2}$ | $5.21 \times 10^{2}$ | $5.21 \times 10^{2}$ | $5.20 \times 10^{2}$ |
| | Std | $9.69 \times 10^{2}$ | $8.62 \times 10^{-2}$ | $1.14 \times 10^{1}$ | $4.43 \times 10^{-2}$ | $3.13 \times 10^{2}$ | $1.06 \times 10^{-1}$ |
| $F_{19}$ | Mean | $1.39 \times 10^{9}$ | $4.83 \times 10^{5}$ | $2.96 \times 10^{3}$ | $5.07 \times 10^{4}$ | $2.27 \times 10^{3}$ | $1.98 \times 10^{3}$ |
| | Std | $1.98 \times 10^{4}$ | $4.22 \times 10^{5}$ | $2.00 \times 10^{3}$ | $2.89 \times 10^{4}$ | $2.91 \times 10^{3}$ | $1.56 \times 10^{3}$ |
| $F_{20}$ | Mean | $3.17 \times 10^{3}$ | $3.04 \times 10^{5}$ | $3.13 \times 10^{3}$ | $8.77 \times 10^{5}$ | $4.66 \times 10^{3}$ | $2.67 \times 10^{3}$ |
| | Std | $5.98 \times 10^{2}$ | $2.31 \times 10^{5}$ | $1.56 \times 10^{1}$ | $3.70 \times 10^{5}$ | $4.86 \times 10^{3}$ | $2.25 \times 10^{2}$ |
| $F_{21}$ | Mean | $9.39 \times 10^{8}$ | $1.12 \times 10^{7}$ | $3.56 \times 10^{4}$ | $5.20 \times 10^{6}$ | $4.10 \times 10^{4}$ | $6.27 \times 10^{3}$ |
| | Std | $1.14 \times 10^{6}$ | $5.43 \times 10^{6}$ | $1.05 \times 10^{5}$ | $2.42 \times 10^{6}$ | $2.33 \times 10^{5}$ | $6.06 \times 10^{4}$ |
| $F_{22}$ | Mean | $3.20 \times 10^{3}$ | $3.79 \times 10^{5}$ | $1.26 \times 10^{4}$ | $4.00 \times 10^{3}$ | $3.20 \times 10^{3}$ | $3.20 \times 10^{3}$ |
| | Std | $7.83 \times 10^{-4}$ | $2.41 \times 10^{5}$ | $9.53 \times 10^{0}$ | $2.93 \times 10^{2}$ | $1.92 \times 10^{3}$ | 0 |

As can be seen from Table 3, in low-dimensional functions, the optimization accuracy of CLTSO is only slightly weaker than its competitors in $F_6$, $F_8$, $F_{12}$, and $F_{13}$. Among the remaining 10 benchmark functions, CLTSO not only has significantly better solution accuracy than its competitors, but also has better robustness. This shows that the Circle chaotic map operator can help CLTSO obtain more diverse candidate solutions, and each candidate solution can continuously update and finally select the optimal solution during the iteration.

When the dimension of the benchmark function is 100, CLTSO has better optimization performance in dealing with higher dimensional and more complex problems. Only in the $F_8$ test function is the optimization accuracy of CLTSO slightly worse than that of CMA-ES. In the remaining 12 functions, CLTSO has the best optimization accuracy, and CLTSO can find the theoretical optimal value in $F_1$, $F_2$, $F_3$, $F_4$, $F_9$, $F_{10}$, and $F_{11}$. From the robustness of the algorithm, CLTSO obtains the minimum Std value in all benchmark functions, which indicates that CLTSO has more stable exploration ability than other competitors. This is due to the fact that the Circle chaotic map strategy helps CLTSO to obtain a richer population diversity, which allows the initial tuna to be evenly distributed in the search space. In addition, during the execution of CLTSO, the Levy flight operator strengthens the exploration capability of the algorithm, and the nonlinear adaptive weight operator can well balance the exploration and exploitation capability of CLTSO.

The experimental results of the CEC2014 function indicate that all algorithms do not obtain the theoretical optimal value, but CLTSO can still achieve more excellent optimization accuracy than other competitors in $F_{16}$~$F_{22}$. This effectively proves that the improved nonlinear tuna swarm optimization algorithm based on the Circle chaotic map strategy and the Levy flight operator can adapt to more complex and challenging optimization problems.

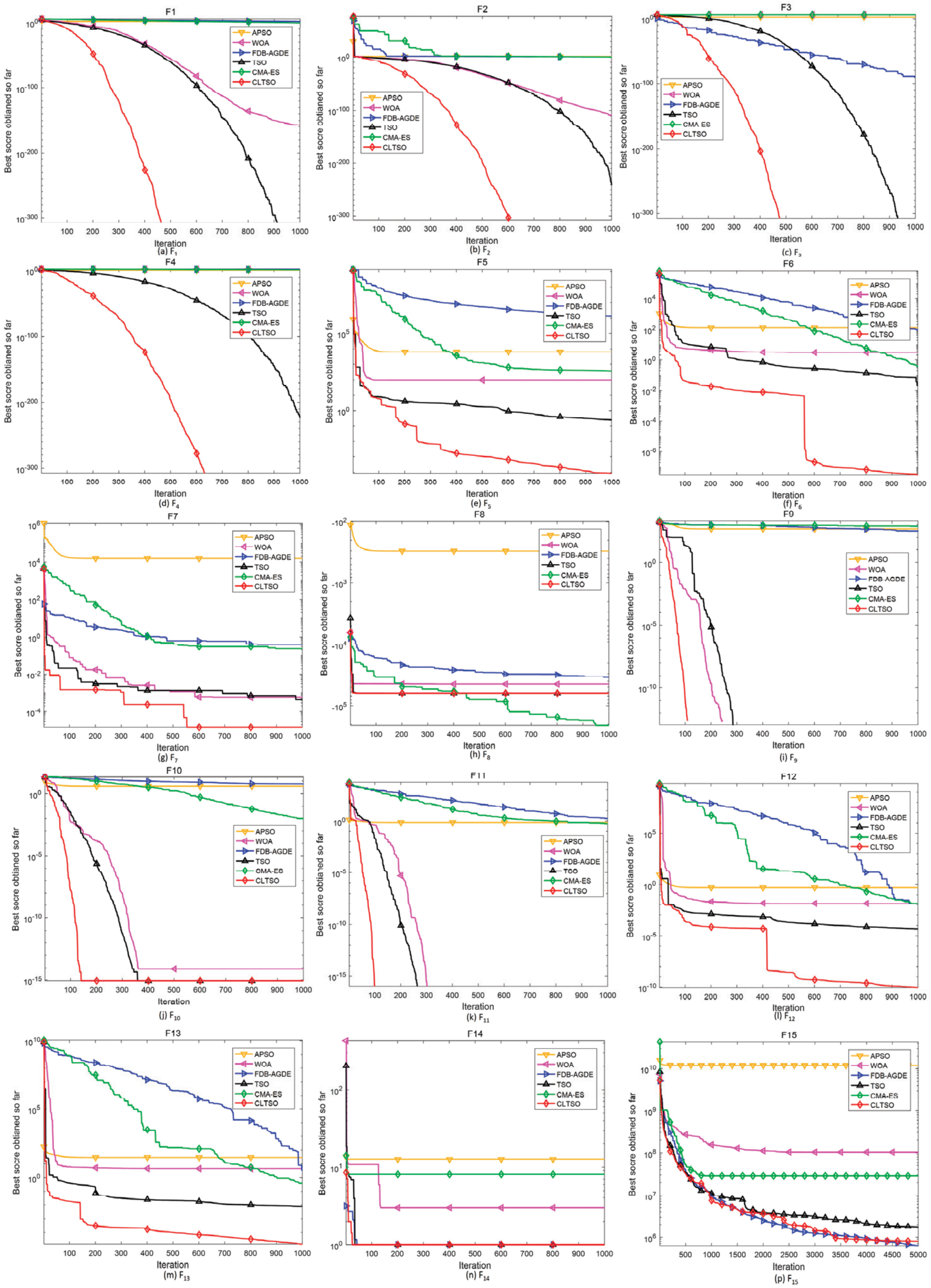
To more intuitively observe the convergence ability of CLTSO and the competitors, Figure 7 draw their operating curves. The images of $F_1$~$F_{13}$ are drawn in 100 dimensions, the image of $F_{14}$ is drawn in its suitable dimension, and the images of $F_{15}$~$F_{22}$ are drawn in 50 dimensions.

The convergence curves of these algorithms indicate that CLTSO has a better convergence performance than the competitors. For simple optimization problems, CLTSO can obtain theoretical optimal values within 500~600 iterations. For complex and challenging problems, CLTSO can also maintain a faster convergence rate and get rid of the influence of local attraction points, and ultimately achieve higher optimization accuracy.

In order to further show whether CLTSO has obvious advantage over other algorithms, this paper uses the Wilcoxon [56] statistical method and the Friedman method to analyze the experimental results of these algorithms in 100-dimensional benchmark functions. The results of $F_{14}$ is based on its suitable dimensions. The experimental data of eight CEC2014 benchmark functions are measured in 50 dimensions. The results of the Friedman test and the *p*-value of the Wilcoxon test are listed in Tables 7 and 8, respectively.

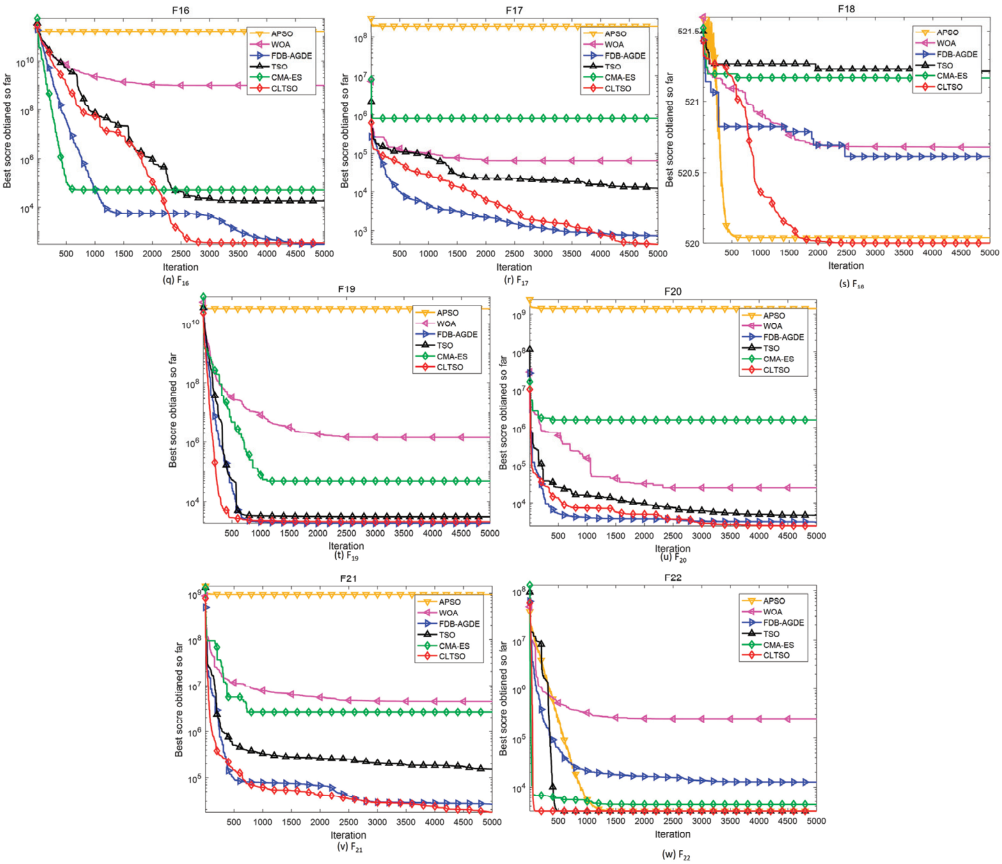
**Table 7.** Results of Friedman test.

| Algorithm | Rank Mean |
|---|---|
| CLTSO | 1.39 |
| TSO | 2.48 |
| FDB-AGDE | 3.68 |
| CMA-ES | 4.09 |
| WOA | 4.14 |
| APSO | 5.23 |



(a) $F_1$
(b) $F_2$
(c) $F_3$
(d) $F_4$
(e) $F_5$
(f) $F_6$
(g) $F_7$
(h) $F_8$
(i) $F_9$
(j) $F_{10}$
(k) $F_{11}$
(l) $F_{12}$
(m) $F_{13}$
(n) $F_{14}$
(p) $F_{15}$

**Figure 7.** *Cont.*



(q) $F_{16}$ (r) $F_{17}$ (s) $F_{18}$ (t) $F_{19}$ (u) $F_{20}$ (v) $F_{21}$ (w) $F_{22}$

**Figure 7.** Convergence curve of each algorithm (F1~F22).

The Friedman test is a nonparametric statistical analysis method, which uses rank mean to test whether there are significant differences in multiple population distributions. Because the problem in this paper is to find the minimum value, a smaller rank mean value in the Friedman test results indicates better performance of the algorithm. As can be seen from Table 6, CLTSO has the smallest rank mean and TSO ranks the second, followed by CMA-ES, WOA, DE, and APSO.

In the Wilcoxon statistical test results, if the *p*-value is less than 0.05 and close to 0, this indicates that the experimental results of the two algorithms are significantly different. If the *p*-value exceeds 0.05, this indicates that the experimental results of the two algorithms are not significantly different. If the *p*-value is equal to NaN, this means that the experimental results of the two algorithms are not different. As can be seen from Table 7, except for the last column, the *p*-values of CLTSO are basically less than 0.05 and close to 0, which indicates that CLTSO has significant advantages compared with other algorithms. It is not difficult to find that half of the *p*-values for Wilcoxon analysis of CLTSO vs. TSO are greater than 0.05. This is because both CLTSO and TSO can find the theoretical optimal value in these functions, or the optimal value found by TSO is not much different from that found by CLTSO. From the optimization curves of TSO and CLTSO, we can see that although the calculation results of these two algorithms are not very different in those functions with *p*-values greater than 0.05, the speed of CLTSO is generally much faster than that of TSO.

**Table 8.** Results of Wilcoxon test.

| Function | CLTSO vs. WOA | CLTSO vs. APSO | CLTSO vs. FDB-AGDE | CLTSO vs. CMAES | CLTSO vs. TSO |
|---|---|---|---|---|---|
| $F_1$ | $1.21 \times 10^{-12}$ | $1.21 \times 10^{-12}$ | $7.94 \times 10^{-3}$ | $1.21 \times 10^{-12}$ | NaN |
| $F_2$ | $1.21 \times 10^{-12}$ | $1.21 \times 10^{-12}$ | $7.94 \times 10^{-3}$ | $1.21 \times 10^{-12}$ | $1.21 \times 10^{-12}$ |
| $F_3$ | $1.21 \times 10^{-12}$ | $1.21 \times 10^{-12}$ | $7.94 \times 10^{-3}$ | $1.21 \times 10^{-12}$ | NaN |
| $F_4$ | $1.21 \times 10^{-12}$ | $1.21 \times 10^{-12}$ | $7.94 \times 10^{-3}$ | $1.21 \times 10^{-12}$ | $1.21 \times 10^{-12}$ |
| $F_5$ | $3.02 \times 10^{-11}$ | $3.02 \times 10^{-11}$ | $7.94 \times 10^{-3}$ | $3.02 \times 10^{-11}$ | $4.18 \times 10^{-9}$ |
| $F_6$ | $3.02 \times 10^{-11}$ | $3.02 \times 10^{-11}$ | $7.94 \times 10^{-3}$ | $3.02 \times 10^{-11}$ | $2.78 \times 10^{-7}$ |
| $F_7$ | $3.82 \times 10^{-10}$ | $3.02 \times 10^{-11}$ | $7.94 \times 10^{-3}$ | $3.02 \times 10^{-11}$ | $6.20 \times 10^{-4}$ |
| $F_8$ | $3.02 \times 10^{-11}$ | $3.02 \times 10^{-11}$ | $7.94 \times 10^{-3}$ | $3.02 \times 10^{-11}$ | $3.65 \times 10^{-8}$ |
| $F_9$ | NaN | $1.21 \times 10^{-12}$ | $7.94 \times 10^{-3}$ | $1.21 \times 10^{-12}$ | NaN |
| $F_{10}$ | $3.06 \times 10^{-9}$ | $1.21 \times 10^{-12}$ | $7.94 \times 10^{-3}$ | $1.21 \times 10^{-12}$ | NaN |
| $F_{11}$ | NaN | $1.21 \times 10^{-12}$ | $7.94 \times 10^{-3}$ | $1.21 \times 10^{-12}$ | NaN |
| $F_{12}$ | $3.02 \times 10^{-11}$ | $3.02 \times 10^{-11}$ | $7.94 \times 10^{-3}$ | $3.02 \times 10^{-11}$ | $6.53 \times 10^{-8}$ |
| $F_{13}$ | $3.02 \times 10^{-11}$ | $3.02 \times 10^{-11}$ | $7.94 \times 10^{-3}$ | $3.02 \times 10^{-11}$ | $1.69 \times 10^{-9}$ |
| $F_{14}$ | $1.57 \times 10^{-11}$ | $1.39 \times 10^{-4}$ | NaN | $1.57 \times 10^{-11}$ | $1.22 \times 10^{-1}$ |
| $F_{15}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $1.59 \times 10^{-2}$ | $7.94 \times 10^{-3}$ | $1.51 \times 10^{-1}$ |
| $F_{16}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $8.41 \times 10^{-1}$ | $4.21 \times 10^{-1}$ | $6.90 \times 10^{-1}$ |
| $F_{17}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ |
| $F_{18}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ |
| $F_{19}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $8.41 \times 10^{-1}$ | $7.94 \times 10^{-3}$ | $8.41 \times 10^{-1}$ |
| $F_{20}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ |
| $F_{21}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $4.21 \times 10^{-1}$ | $7.94 \times 10^{-3}$ | $1.51 \times 10^{-1}$ |
| $F_{22}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $7.94 \times 10^{-3}$ | $1.00 \times 10^{0}$ |

Finally, this paper quantitatively analyzes all the algorithms in the experiment. The quantitative analysis of these algorithms is based on the mean absolute error (MAE) of 22 benchmark functions. In mathematics, MAE is a measure of the error between paired observations expressing the same phenomenon. The mathematical model of MAE is as follows:

$$MAE = \frac{\sum_{i=1}^{N} |m_i - o_i|}{N} \tag{26}$$

where $N$ is the total amount of benchmark functions used for testing, $m_i$ is the average of the optimal results calculated by the algorithm, and $o_i$ is the theoretical optimal value of the $i$th benchmark function.

Table 9 shows the MAE ranking results of these algorithms. The MAE value of CLTSO ranks the first among all competitors, and FDB-AGDE ranks the second. The above data intuitively illustrate the advantage of CLTSO.

**Table 9.** MAE ranking results of each algorithm.

| Algorithm | MAE |
|---|---|
| CLTSO | $2.06 \times 10^{4}$ |
| FDB-AGDE | $2.70 \times 10^{4}$ |
| CMA-ES | $1.21 \times 10^{6}$ |
| TSO | $3.82 \times 10^{6}$ |
| WOA | $3.55 \times 10^{9}$ |
| APSO | $9.60 \times 10^{9}$ |

The time consumed by these algorithms in functions $F_1$~$F_{22}$ are shown in Table 10. The numerical unit is second. The analysis of the time they consumed indicates that the time complexity of CLTSO is slightly higher than that of TSO, but the increase is trivial. The improved operator proposed in this paper only increases the time complexity a little but greatly enhances the optimization performance of the CLTSO algorithm.

**Table 10.** The execution time of each algorithm.

| Function | APSO | WOA | FDB-AGDE | CMA-ES | TSO | CLTSO |
|---|---|---|---|---|---|---|
| $F_1$ | 0.5014 | 0.2206 | 8.2466 | 2.6347 | 0.2277 | 0.2603 |
| $F_2$ | 0.4465 | 0.3333 | 8.7963 | 2.8552 | 0.2389 | 0.27260 |
| $F_3$ | 0.5782 | 1.3826 | 8.4424 | 4.0751 | 1.2743 | 1.2819 |
| $F_4$ | 4.3284 | 0.2083 | 8.8472 | 2.4857 | 0.1960 | 0.1942 |
| $F_5$ | 0.7193 | 0.2458 | 3.586 | 2.7163 | 0.2440 | 0.3442 |
| $F_6$ | 0.8572 | 0.1921 | 8.8967 | 2.4867 | 0.1879 | 0.1946 |
| $F_7$ | 0.7557 | 0.4226 | 13.6329 | 2.7073 | 0.3906 | 0.3860 |
| $F_8$ | 1.0016 | 0.2599 | 23.6696 | 2.6709 | 0.2689 | 0.2698 |
| $F_9$ | 0.5113 | 0.2074 | 26.1973 | 2.5678 | 0.2126 | 0.2246 |
| $F_{10}$ | 0.5619 | 0.2370 | 21.3863 | 3.9514 | 0.3425 | 0.3350 |
| $F_{11}$ | 0.5321 | 0.4213 | 20.6873 | 2.9784 | 0.3491 | 0.4298 |
| $F_{12}$ | 1.9510 | 0.9889 | 19.5429 | 3.5528 | 0.8823 | 0.8381 |
| $F_{13}$ | 1.8874 | 0.8468 | 11.6196 | 6.1577 | 2.2185 | 1.5804 |
| $F_{14}$ | 2.0775 | 3.0364 | 4.9752 | 4.7462 | 2.2598 | 2.2449 |
| $F_{15}$ | 1.9327 | 2.3167 | 15.5858 | 21.6152 | 2.4425 | 2.6542 |
| $F_{16}$ | 1.4227 | 1.9362 | 14.8224 | 21.4082 | 1.9444 | 2.0605 |
| $F_{17}$ | 1.4958 | 2.0395 | 14.6579 | 22.1484 | 2.0020 | 1.9974 |
| $F_{18}$ | 1.2775 | 2.2311 | 15.6855 | 21.5255 | 2.1373 | 2.2258 |
| $F_{19}$ | 1.7849 | 2.1867 | 15.3276 | 22.2718 | 2.0847 | 2.2293 |
| $F_{20}$ | 2.5258 | 2.2063 | 29.3849 | 20.9746 | 2.1806 | 2.2839 |
| $F_{21}$ | 3.0010 | 2.4824 | 30.1273 | 23.3055 | 2.4337 | 2.5681 |
| $F_{22}$ | 6.1130 | 6.2426 | 49.8559 | 27.5203 | 5.9850 | 6.0176 |

### 4.4. Effectiveness Analysis of Improved Operators

This paper makes three improvements to the original tuna swarm optimization algorithm. Firstly, the improved Circle chaotic mapping strategy is introduced in the initialization phase, which expands the swarm diversity. Secondly, the Levy operator is introduced in the position update phase, which strengthens the global swimming ability of tuna. Finally, the nonlinear adaptive weight strategy is introduced in the TSO iteration stage, which can effectively balance the exploration and the exploitation capabilities of the tuna swarm. Section 3 of this chapter proves that the proposed operator significantly improves the optimization performance of TSO. In addition, to verify the effectiveness of the improvements proposed in this paper, we selected the tuna swarm optimization algorithm (TSO), the improved TSO based on the Levy flight operator (LTSO), the improved TSO based on the Circle chaotic map and nonlinear adaptive weights (CTSO), and CLTSO to conduct a set of comparative experiments. Functions $F_1$~$F_{22}$ are used to test these algorithms in this section, and each algorithm runs 30 times independently. $F_1$~$F_{13}$ are experiments in 100 dimensions, $F_{15}$~$F_{22}$ are experiments in 50 dimensions.

The experimental results of various versions of the improved tuna swarm optimization algorithm are displayed in Table 11. Their convergence curves are displayed in Figure 8.

**Table 11.** Experimental results of various versions of the improved TSO.

| Function | Performance | TSO | LTSO | CTSO | CLTSO |
|---|---|---|---|---|---|
| $F_1$ | Mean | 0 | 0 | 0 | 0 |
| | Std | 0 | 0 | 0 | 0 |
| $F_2$ | Mean | $9.66 \times 10^{-237}$ | 0 | 0 | 0 |
| | Std | 0 | 0 | 0 | 0 |
| $F_3$ | Mean | 0 | 0 | 0 | 0 |
| | Std | 0 | 0 | 0 | 0 |
| $F_4$ | Mean | $1.26 \times 10^{-233}$ | 0 | 0 | 0 |
| | Std | 0 | 0 | 0 | 0 |

**Table 11.** *Cont.*

| Function | Performance | TSO | LTSO | CTSO | CLTSO |
|---|---|---|---|---|---|
| $F_5$ | Mean | $2.52 \times 10^{-3}$ | $4.84 \times 10^{-4}$ | $2.35 \times 10^{-3}$ | $1.27 \times 10^{-5}$ |
| | Std | $3.37 \times 10^{-4}$ | $2.47 \times 10^{-2}$ | $6.09 \times 10^{-3}$ | $5.71 \times 10^{-5}$ |
| $F_6$ | Mean | $9.33 \times 10^{-4}$ | $6.07 \times 10^{-5}$ | $2.92 \times 10^{-4}$ | $3.07 \times 10^{-5}$ |
| | Std | $2.93 \times 10^{-3}$ | $8.82 \times 10^{-5}$ | $1.76 \times 10^{-2}$ | $2.78 \times 10^{-5}$ |
| $F_7$ | Mean | $1.34 \times 10^{-4}$ | $4.20 \times 10^{-5}$ | $8.46 \times 10^{-5}$ | $4.07 \times 10^{-5}$ |
| | Std | $1.62 \times 10^{-4}$ | $6.46 \times 10^{-5}$ | $1.41 \times 10^{-4}$ | $3.07 \times 10^{-5}$ |
| $F_8$ | Mean | $-4.19 \times 10^{4}$ | $-4.19 \times 10^{4}$ | $-4.19 \times 10^{4}$ | $-4.19 \times 10^{4}$ |
| | Std | $2.51 \times 10^{4}$ | $2.51 \times 10^{4}$ | $7.54 \times 10^{3}$ | $5.30 \times 10^{-8}$ |
| $F_9$ | Mean | 0 | 0 | 0 | 0 |
| | Std | 0 | 0 | 0 | 0 |
| $F_{10}$ | Mean | $8.88 \times 10^{-16}$ | $8.88 \times 10^{-16}$ | $8.88 \times 10^{-16}$ | $8.88 \times 10^{-16}$ |
| | Std | 0 | 0 | 0 | 0 |
| $F_{11}$ | Mean | 0 | 0 | 0 | 0 |
| | Std | 0 | 0 | 0 | 0 |
| $F_{12}$ | Mean | $5.26 \times 10^{-5}$ | $1.41 \times 10^{-6}$ | $7.65 \times 10^{-7}$ | $5.14 \times 10^{-10}$ |
| | Std | $2.72 \times 10^{-5}$ | $2.04 \times 10^{-7}$ | $8.98 \times 10^{-5}$ | $1.52 \times 10^{-7}$ |
| $F_{13}$ | Mean | $7.86 \times 10^{-4}$ | $7.47 \times 10^{-6}$ | $9.84 \times 10^{-6}$ | $3.60 \times 10^{-7}$ |
| | Std | $7.73 \times 10^{-4}$ | $5.16 \times 10^{-3}$ | $1.58 \times 10^{-3}$ | $2.08 \times 10^{-5}$ |
| $F_{14}$ | Mean | $9.98 \times 10^{-1}$ | $9.98 \times 10^{-1}$ | $9.98 \times 10^{-1}$ | $9.98 \times 10^{-1}$ |
| | Std | $5.99 \times 10^{-1}$ | $5.99 \times 10^{-1}$ | $5.99 \times 10^{-1}$ | $1.72 \times 10^{-16}$ |
| $F_{15}$ | Mean | $1.15 \times 10^{6}$ | $1.69 \times 10^{6}$ | $2.32 \times 10^{6}$ | $9.51 \times 10^{5}$ |
| | Std | $1.45 \times 10^{6}$ | $6.65 \times 10^{5}$ | $2.06 \times 10^{6}$ | $3.35 \times 10^{5}$ |
| $F_{16}$ | Mean | $1.38 \times 10^{4}$ | $1.12 \times 10^{4}$ | $1.61 \times 10^{3}$ | $3.40 \times 10^{2}$ |
| | Std | $7.08 \times 10^{3}$ | $1.93 \times 10^{3}$ | $8.20 \times 10^{3}$ | $2.05 \times 10^{3}$ |
| $F_{17}$ | Mean | $3.16 \times 10^{3}$ | $4.72 \times 10^{2}$ | $4.57 \times 10^{3}$ | $3.71 \times 10^{2}$ |
| | Std | $6.55 \times 10^{3}$ | $2.28 \times 10^{2}$ | $1.62 \times 10^{4}$ | $4.36 \times 10^{1}$ |
| $F_{18}$ | Mean | $5.21 \times 10^{2}$ | $5.20 \times 10^{2}$ | $5.21 \times 10^{2}$ | $5.20 \times 10^{2}$ |
| | Std | $3.13 \times 10^{2}$ | $3.12 \times 10^{2}$ | $3.13 \times 10^{2}$ | $4.47 \times 10^{-2}$ |
| $F_{19}$ | Mean | $5.54 \times 10^{3}$ | $8.74 \times 10^{3}$ | $4.22 \times 10^{3}$ | $3.78 \times 10^{3}$ |
| | Std | $2.93 \times 10^{3}$ | $3.42 \times 10^{3}$ | $2.35 \times 10^{3}$ | $1.34 \times 10^{3}$ |
| $F_{20}$ | Mean | $6.52 \times 10^{3}$ | $3.09 \times 10^{3}$ | $5.49 \times 10^{3}$ | $2.62 \times 10^{3}$ |
| | Std | $3.39 \times 10^{3}$ | $1.59 \times 10^{3}$ | $4.34 \times 10^{3}$ | $1.85 \times 10^{2}$ |
| $F_{21}$ | Mean | $2.15 \times 10^{4}$ | $7.56 \times 10^{4}$ | $1.95 \times 10^{4}$ | $1.90 \times 10^{4}$ |
| | Std | $1.30 \times 10^{5}$ | $3.01 \times 10^{4}$ | $6.81 \times 10^{4}$ | $3.08 \times 10^{4}$ |
| $F_{22}$ | Mean | $3.20 \times 10^{3}$ | $3.20 \times 10^{3}$ | $3.20 \times 10^{3}$ | $3.20 \times 10^{3}$ |
| | Std | 0 | 0 | 0 | 0 |

As can be seen from Table 10 and Figure 8, CLTSO has higher optimization accuracy than the competitors. In benchmark functions $F_1$, $F_2$, $F_3$, $F_4$, $F_9$, $F_{10}$, $F_{11}$, and $F_{14}$, CLTSO, CTSO, and LTSO can calculate the theoretical optimal values, but CLTSO converges much faster than CTSO and LTSO. The above data indicate that the optimization performance of CTSO and LTSO is more enhanced than the original tuna swarm optimization algorithm, which further confirms the validity of the three modified operators in CLTSO. To demonstrate that the optimization capability of CLTSO is greatly enhanced compared to CTSO and LTSO, Friedman statistical analysis and MAE ranking are conducted based on the data in Table 11. The analysis and ranking results are listed in Tables 12 and 13.

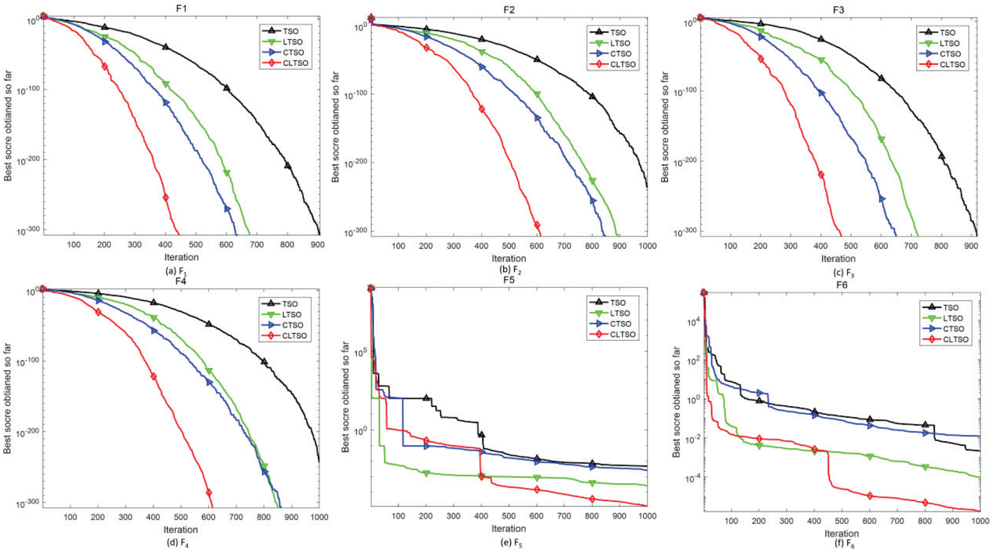
**Table 12.** Results of Friedman statistical analysis.

| Algorithm | Rank Mean |
|---|---|
| CLTSO | 1.80 |
| LTSO | 2.25 |
| CTSO | 2.84 |
| TSO | 3.11 |

**Table 13.** MAE ranking results.

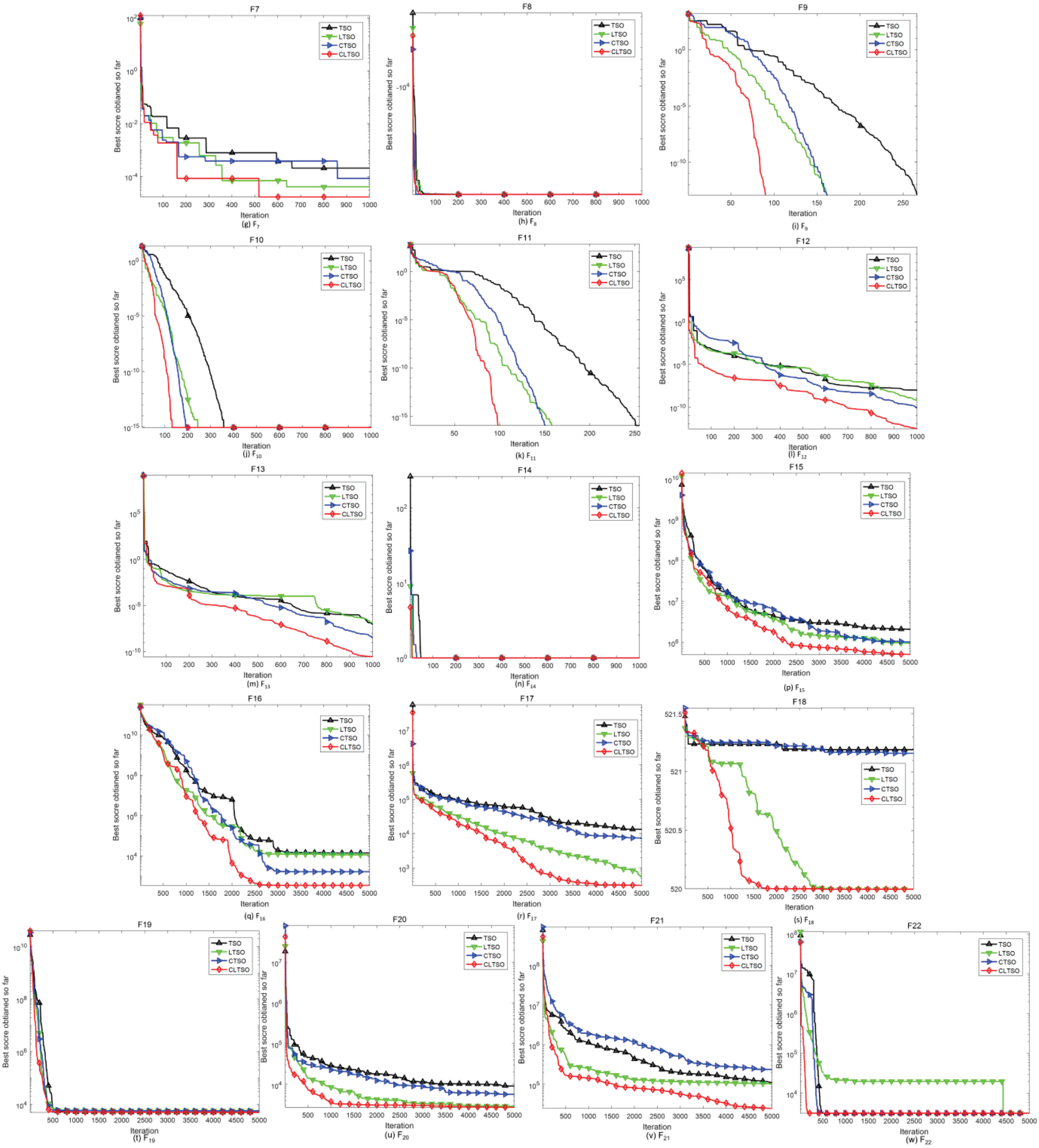
| Algorithm | MAE |
|---|---|
| CLTSO | $4.42 \times 10^4$ |
| LTSO | $8.06 \times 10^4$ |
| CTSO | $1.08 \times 10^5$ |
| TSO | $1.18 \times 10^5$ |

According to the above two tables, it is clear that CLTSO has the smallest rank mean in the Friedman analysis test, LTSO ranks the second, and CTSO ranks the third, followed by TSO. According to the MAE value of each algorithm, CLTSO ranks the first. The above ranking shows that CLTSO can better approximate the theoretical optimal value when dealing with optimization problems. CLTSO has shown much better performance than the competitors. Therefore, the above data and analysis results confirm that the three improved operators proposed in this paper are effective.



(a) $F_1$ (b) $F_2$ (c) $F_3$ (d) $F_4$ (e) $F_5$ (f) $F_6$

**Figure 8.** *Cont.*



(g) $F_7$ (h) $F_8$ (i) $F_9$ (j) $F_{10}$ (k) $F_{11}$ (l) $F_{12}$ (m) $F_{13}$ (n) $F_{14}$ (p) $F_{15}$ (q) $F_{16}$ (r) $F_{17}$ (s) $F_{18}$ (t) $F_{19}$ (u) $F_{20}$ (v) $F_{21}$ (w) $F_{22}$

**Figure 8.** Convergence curves of each version of improved TSO.

## 5. Optimization Engineering Example Using CLTSO

The original intention of meta-heuristic algorithms is to optimize the engineering problems encountered. How to improve the precision of engineering practice is the concern of researchers. To verify the effectiveness of CLTSO for real engineering problems, CLTSO is applied to the modification design of a BP neural network. The BP neural network is a model proposed by McCulloch to train the network based on error back propagation. It is one of the most mature and widely used artificial neural network modules. The BP neural network is widely used in pattern recognition, classification and prediction, nonlinear modeling, etc. Figure 9 shows a BP neural network topology with $d$ input neurons, $l$ output neurons, and $q$ hidden layer neurons.

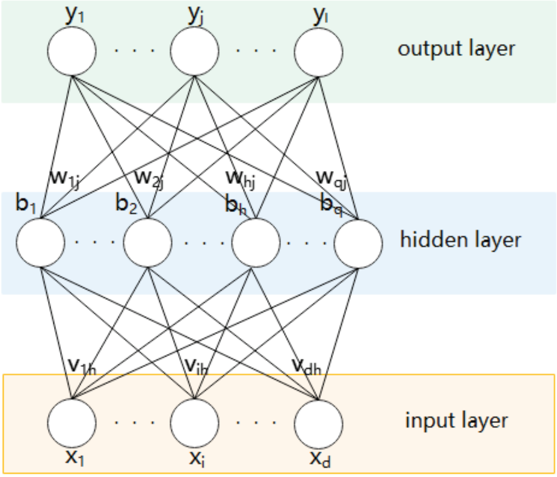


**Figure 9.** BP neural network topology.

$v_{ih}$ is the weight between the $i$th node in the input layer and the $h$th node in the hidden layer. $w_{hj}$ is the weight between the $h$th node in the input layer and the $j$th node in the hidden layer. The threshold of the $j$th node in the output layer is expressed by $\theta_j$. Therefore, the input value received by the hidden layer $h$th neurons in the network model is as follows:

$$\alpha_h = \sum\nolimits_{i=1}^{d} v_{ih} x_i \tag{27}$$

The value received by the $j$th node in the output layer is as follows:

$$\beta_j = \sum\nolimits_{h=1}^{q} w_{ih} b_h \tag{28}$$

where $b_h$ is the output value of the $h$th neuron in the hidden layer. Taking training case ($x_k$, $y_k$) as an example, we assume that the output of the network model is as follows:

$$\hat{y}_k = \left(\hat{y}_1^k, \hat{y}_2^k, \cdots, \hat{y}_l^k\right) \tag{29}$$

$$\hat{y}_j^k = f(\beta_j - \theta_j) \tag{30}$$

Therefore, the mean square error of the network on example ($x_k$, $y_k$) is as follows:

$$E_k = 1/2 \sum\nolimits_{j=1}^{l} \left(\hat{y}_j^k - y_j^k\right)^2 \tag{31}$$

where n is the total amount of training samples, m is the total amount of input nodes, $x_i^k$ is the output value of the network model, and $d_i^k$ is the real value of training samples.

In the training process of the model, the error will be transmitted back to the hidden nodes. The model will adjust the weights and thresholds between each layer of nodes based on the error, and finally make the error achieve satisfactory accuracy. At present, the training methods of the BP neural network are mostly gradient descent. The training accuracy of the network model is extremely sensitive to the initial weight value and the learning rate. Therefore, when the objective function has multiple extreme values, the neural network is easily attracted by local extreme values. This will lead to a serious degradation in the performance of the algorithm. In order to optimize the performance of the BP network model and verify the optimization ability of CLTSO, a CLTSO-BP neural network model is proposed. The basic idea of the model is to use the weights and thresholds of each node in the BP model as the tuna individual in the CLTSO algorithm and use MSE as the fitness function in the CLTSO algorithm. CLTSO optimizes the MSE of the model to obtain the optimized initial value weight and the threshold.

To compare the capability of the CLTSO-BP neural network with the original BP model, three popular datasets from the UCI machine learning and intelligent system center, Iris, Wine, and Wine Quality, are selected to design a comparative experiment. This experiment compares the classification accuracy of the CLTSO-BP neural network model and the BP model on the above three datasets.

In the experiment, the total amount of tuna is 30, the CLTSO algorithm is executed 30 times in total, and the neural network is executed 500 times in total. Table 14 shows the comparison results of the CLTSO-BP neural network model and the original BP model.

**Table 14.** The comparison results of the two models.

| Dataset | Model | Classification Accuracy |
|---|---|---|
| Iris | CLTSO-BP neural network | 100% |
| | BP neural network | 95.2% |
| Wine | CLTSO-BP neural network | 100% |
| | BP neural network | 94.4% |
| Wine Quality | CLTSO-BP neural network | 65.6% |
| | BP neural network | 45.2% |

By comparing the result of the CLTSO-BP neural network and the original BP model on three datasets, it is found that the new model can obtain more ideal classification results. It also indicates that CLTSO can show excellent performance in multi-layer perceptron training difficulties.

## 6. Conclusions

The tuna swarm optimization algorithm is widely recognized by scholars because of its simple structure and low number of parameters. The tuna swarm optimization algorithm has excellent optimization performance, but it can still be further improved. When dealing with simple problems, the solving speed of TSO can still be further improved. When facing complex problems, it is difficult for TSO to escape the attraction of local optimal value. Therefore, this article proposes a modified nonlinear tuna swarm optimization algorithm based on Circle chaotic map and Levy flight operator. The optimization performance of CLTSO has been fully verified in 22 benchmark functions. The results show that CLTSO outperforms the comparable algorithms. Comparation data based on 22 benchmark functions were analyzed using Wilcoxon's test, Friedman's test, and MAE. The analysis conclusion indicates that the rank mean and MAE value of CLTSO are superior to other advanced algorithms such as CMA-ES. Finally, this paper optimizes the BP neural network based on CLTSO. The CLTSO-BP neural network model is tested using three popular datasets from the UCI Machine Learning and Intelligent System Center. Compared with the original BP model, the new model optimizes the classification accuracy. However, for the optimization problem of more complex datasets, the classification ability of the CLTSO-BP neural network still needs to be improved. Possible directions include increasing the swarm size of the algorithm and the total number of CLTSO operations to obtain a higher quality solution, which is also the target of continuous research in the future. In addition, the advantages of CLTSO in solving some complex multimodal functions can still be further improved, which is also one of the key research directions in the future. CLTSO has the advantages of fast convergence and high convergence accuracy, which can be applied in practical projects such as workshop scheduling and distribution network reconstruction.

**Author Contributions:** Conceptualization, W.W. and J.T.; methodology, W.W.; software, W.W.; validation, W.W.; formal analysis, W.W.; investigation, W.W.; resources, W.W.; data curation, W.W.; writing—original draft preparation, W.W.; writing—review and editing, W.W. and J.T.; visualization, W.W.; supervision, J.T.; project administration, J.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**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. Narendra, P.M.; Fukunaga, K. A branch and bound algorithm for feature subsets election. *IEEE Trans Comput.* **1977**, *26*, 917–922. [CrossRef]
2. Wu, G.; Pedrycz, W.; Suganthan, P.N.; Mallipeddi, R. A variable reduction strategy for evolutionary algorithms handling equality constraints. *Appl. Soft Comput. J.* **2015**, *37*, 774–786. [CrossRef]
3. Zhang, H.; Cui, L.; Zhang, X.; Luo, Y. Data-driven robust approximate optimal tracking control for unknown general non-linear systems using adaptive dynamic programming method. *IEEE Trans. Neural Netw.* **2011**, *22*, 2226–2236. [CrossRef]
4. Slowik, A.; Halina, K. Nature inspired methods and their industry applications—Swarm intelligence algorithms. *IEEE Trans. Ind. Inform.* **2017**, *14*, 1004–1015. [CrossRef]
5. Chakraborty, A.; Kar, A.K. Swarm intelligence: A review of algorithms. *Nat.-Inspir. Comput. Optim.* **2017**, *10*, 475–494.
6. Liu, W.; Dridi, M.; Fei, H.; el Hassani, A.H. Hybrid metaheuristics for solving a home health care routing and scheduling problem with time windows, synchronized visits and lunch breaks. *Expert Syst. Appl.* **2021**, *183*, 115307. [CrossRef]
7. Wang, X.; Zhao, H.; Han, T.; Zhou, H.; Li, C. A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem. *Appl. Soft Comput.* **2019**, *78*, 240–260. [CrossRef]
8. Wang, Y.-G. A maximum-likelihood method for estimating natural mortality and catchability coefficient from catch-and-effort data. *Mar. Freshw. Res.* **1999**, *50*, 307–311. [CrossRef]
9. Wu, J.; Ding, Z. Improved grey model by dragonfly algorithm for Chinese tourism demand forecasting. In Proceedings of the 33rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Kitakyushu, Japan, 22–25 September 2020.
10. Wu, J.; Cui, Z.; Chen, Y.; Kong, D.; Wang, Y.-G. A new hybrid model to predict the electrical load in five states of Australia. *Energy* **2019**, *166*, 598–609. [CrossRef]
11. Webb, B. Swarm Intelligence: From Natural to Artificial Systems. *Connect. Sci.* **2002**, *14*, 163–164. [CrossRef]
12. Kennedy, J. Swarm Intelligence. In *Handbook of Nature-Inspired and Innovative Computing*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 187–219.
13. Ashlock, D. *Evolutionary Computation for Modeling and Optimization*; Springer: New York, NY, USA, 2006; Volume 51, p. 743.
14. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput.-Aided Des.* **2011**, *43*, 303–315. [CrossRef]
15. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2015**, *27*, 495–513. [CrossRef]
16. Chopra, N.; Ansari, M.M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Syst. Appl.* **2022**, *198*, 116924. [CrossRef]
17. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
18. Chen, D.; Ge, Y.; Wan, Y.; Deng, Y.; Chen, Y.; Zou, F. Poplar optimization algorithm: A new meta-heuristic optimization technique for numerical optimization and image segmentation. *Expert Syst. Appl.* **2022**, *200*, 1–17. [CrossRef]
19. Man, K.F.; Tang, K.S.; Kwong, S. Genetic Algorithms. *Perspect. Neural Comput.* **1989**, *83*, 55–80.
20. Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [CrossRef]
21. Du, Z.; Li, S.; Sun, Y.; Li, N. Adaptive particle swarm optimization algorithm based on levy flights mechanism. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017.
22. Yu, H.; Yu, Y.; Liu, Y.; Wang, Y.; Gao, S. Chaotic grey wolf optimization. In Proceedings of the 2016 International Conference on Progress in Informatics and Computing (PIC), Beijing, China, 23–26 December 2016; pp. 103–113.
23. Yuan, X.; Yang, D.; Liu, H. MPPT of PV system under partial shading condition based on adaptive inertia weight particle swarm optimization algorithm. In Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, China, 8–12 June 2015; pp. 729–733.
24. Park, S.; Kim, Y.; Kim, J.; Lee, J. Speeded-up cuckoo search using opposition-based learning. In Proceedings of the 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), Seoul, Korea, 22–25 October 2014; pp. 535–539.

25. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **2013**, *222*, 175–184. [CrossRef]
26. Xie, W.; Wang, J.S.; Tao, Y. Improved Black Hole Algorithm Based on Golden Sine Operator and Levy Flight Operator. *IEEE Access* **2019**, *7*, 161459–161486. [CrossRef]
27. Xie, L.; Han, T.; Zhou, H.; Zhang, Z.-R.; Han, B.; Tang, A. Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization. *Comput. Intell. Neurosci.* **2021**, *2021*, 9210050. [CrossRef]
28. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
29. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
30. Hu, D.; Yang, S. Improved Tuna Algorithm to Optimize ELM Model for PV Power Prediction. *J. Wuhan Univ. Technol.* **2022**, *44*, 97–104.
31. Kumar, C.; Mary, D.M. A novel chaotic-driven Tuna Swarm Optimizer with Newton-Raphson method for parameter identification of three-diode equivalent circuit model of solar photovoltaic cells/modules. *Optik* **2022**, *264*, 169379. [CrossRef]
32. Arora, S.; Anand, P. Chaotic grasshopper optimization algorithm for global optimization. *Neural Comput. Appl.* **2019**, *31*, 4385–4405. [CrossRef]
33. Guangyuan, P.; Junfei, Q.; Honggui, H. A new strategy of chaotic PSO and its application in optimization design for pipe network. In Proceedings of the 32nd Chinese Control Conference, Xi'an, China, 26–28 July 2013; pp. 8022–8027.
34. Pluhacek, M.; Senkerik, R.; Zelinka, I.; Davendra, D. Designing PID Controllers by Means of PSO Algorithm Enhanced by Various Chaotic Maps. In Proceedings of the 2013 8th EUROSIM Congress on Modelling and Simulation, Cardiff, UK, 10–13 September 2013; pp. 19–23.
35. Zhao, J. Chaotic particle swarm optimization algorithm based on tent mapping for dynamic origin-destination matrix estimation. In Proceedings of the 2011 International Conference on Electric Information and Control Engineering, Wuhan, China, 15–17 April 2011; pp. 221–224.
36. Zhang, J.; Zhu, Y.; Zhu, H.; Cheng, J. Some improvements to logistic map for chaotic signal generator. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 1090–1093.
37. Li, M.; Sun, X.; Li, W.; Wang, Y. Improved Chaotic Particle Swarm Optimization using circle map for training SVM. In Proceedings of the 2009 Fourth International on Conference on Bio-Inspired Computing, Beijing, China, 16–19 October 2009; pp. 1–7.
38. Vasuyta, K.; Zakharchenko, I. Modified discrete chaotic map bas-ed on Chebyshev polynomial. In Proceedings of the 2016 Third International Scientific-Practical Conference Problems of Info communications Science and Technology (PIC S&T), Kharkov, Ukraine, 4–6 October 2016; pp. 217–219.
39. Jiteurtragool, N.; Ketthong, P.; Wannaboon, C.; San-Um, W. A topologically simple keyed hash function based on circular chaotic sinusoidal map network. In Proceedings of the 2013 15th International Conference on Advanced Communications Technology (ICACT), Seoul, Korea, 27–30 January 2013; pp. 1089–1094.
40. Petavratzis, E.; Moysis, L.; Volos, C.; Nistazakis, H.; Muñoz-Pacheco, J.M.; Stouboulos, I. Motion Control of a Mobile Robot Based on a Chaotic Iterative Map. In Proceedings of the 2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST), Pradesh, India, 7–9 September 2020; pp. 1–4.
41. Zhang, D.M.; Xu, H.; Wang, Y.R.; Song, T.; Wang. Whale optimization algorithm for embedded circle mapping and one-dimensional oppositional learning based small hole imaging. *Control. Decis.* **2021**, *36*, 1173–1180.
42. Song, L.; Chen, W.; Chen, W.; Lin, Y.; Sun, X. Improvement and application of sparrow search algorithm based on hybrid strategy. *J. Beijing Univ. Aeronaut. Astronaut.* **2021**, *1*, 1–16.
43. Viswanathan, G.M.; Afanasyev, V.; Buldyrev, S.V.; Havlin, S.; Da Luz, M.G.E.; Raposo, E.P.; Stanley, H.E. Levy fights search patterns of biological organisms. *Phys. A Stat. Mech. Its Appl.* **2001**, *295*, 85–88. [CrossRef]
44. Biagini, F.; Hu, Y.; Øksendal, B.; Zhang, T. Stochastic Optimal Control and Applications. In *Stochastic Calculus for Fractional Brownian Motion and Applications*; Springer: London, UK, 2008.
45. Yan, X.F.; Ye, D.Y. An improved flora foraging algorithm based on Levy flight. *Comput. Syst. Appl.* **2015**, *24*, 124–132.
46. Haklı, H.; Uğuz, H. A novel particle swarm optimization algorithm with Levy flight. *Appl. Soft Comput.* **2014**, *23*, 333–345. [CrossRef]
47. Liu, C.; Ye, C. Bat algorithm with Levy flight characteristics. *Chin. J. Intell. Syst.* **2013**, *8*, 240–246.
48. Mantegna, R.N. Fast accurate algorithm for numerical simulation of Lévy stable stochastic processes. *Phys. Rev.* **1994**, *49*, 4677–4689. [CrossRef] [PubMed]
49. Zhang, J.; Wang, J.S. Improved Whale Optimization Algorithm Based on Nonlinear Adaptive Weight and Golden Sine Operator. *IEEE Access.* **2020**, *8*, 77013–77048. [CrossRef]
50. Zhang, J.; Wang, J.S. Improved Salp Swarm Algorithm Based on Levy Flight and Sine Cosine Operator. *IEEE Access.* **2020**, *8*, 99740–99771. [CrossRef]
51. Aloui, M.; Hamidi, F. A Chaotic Krill Herd Optimization Algorithm for Global Numerical Estimation of the Attraction. *Domain Nonlinear Syst.* **2021**, *9*, 1743.
52. Liang, J.J.; Qu, B.Y.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*; Technical Report; Computational Intelligence Laboratory, Zhengzhou University: Zhengzhou, China, 2013; p. 201311.

53. Reddy, R.B.; Uttara, K.M. Performance Analysis of Mimo Radar Waveform Using Accelerated Particle Swarm Optimization Algorithm. *Signal Image Process.* **2012**, *3*, 4. [CrossRef]
54. Guvenc, U.; Duman, S.; Kahraman, H.T.; Aras, S.; Katı, M. Fitness-Distance Balance based adaptive guided differential evolution algorithm for security-constrained optimal power flow problem incorporating renewable energy sources. *Appl. Soft Comput.* **2021**, *108*, 107421. [CrossRef]
55. Hansen, N.; Müller, S.D.; Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol. Comput.* **2003**, *11*, 1–18. [CrossRef]
56. García, S.; Molina, D.; Lozano, M.; Herrera, F. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization. *J. Heuristics* **2008**, *15*, 617. [CrossRef]

electronics

MDPI

Article

# Estimating Sound Speed Profile by Combining Satellite Data with In Situ Sea Surface Observations

**Zhenyi Ou [1], Ke Qu [1,*], Yafen Wang [2] and Jianbo Zhou [3]**

[1] College of Electronics and Information Engineering, Guangdong Ocean University, Zhanjiang 524088, China
[2] Unit 91977 of the People's Liberation Army, Beijing 100000, China
[3] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China
[*] Correspondence: quke@gdou.edu.cn

**Abstract:** Given that spatiotemporal measurement of the subsurface profile over a wide range are difficult to obtain, surface observations from satellites are often used to estimate the sound speed profile (SSP). This paper proposes a multisource method based on the self-organizing map (SOM) to improve the estimation of the SSP by merging surface observations with satellite data. Surface observations from the Kuroshio Extension Observatory (KEO) were used to supplement satellite observations (anomalies in the measured sea level and sea surface temperature) to this end. Different combinations of the surface parameters were assessed, their errors were analyzed, and differences between the results before and after the multisource parameters were used are discussed. The proposed method significantly increased the accuracy of estimating the SSP when the parameters obtained from in situ measurements were used, with a root mean square error of 2.18 m/s, less than a third of the error obtained when only satellite observations were used. The proposed method provides a new approach to determining an accurate three-dimensional structure of the sound speed when various surface observations are available.

**Keywords:** Kuroshio Extension Observatory; sound speed profile; self-organizing map

**Citation:** Ou, Z.; Qu, K.; Wang, Y.; Zhou, J. Estimating Sound Speed Profile by Combining Satellite Data with In Situ Sea Surface Observations. *Electronics* **2022**, *11*, 3271. https://doi.org/10.3390/electronics11203271

Academic Editor: Arkaitz Zubiaga

Received: 11 September 2022
Accepted: 10 October 2022
Published: 11 October 2022

## 1. Introduction

The sound speed profile (SSP) plays an important role in civil and military marine applications. As it is a key parameter that determines the characteristics of sound transmission, accurately measuring the SSP is critical for such acoustic applications as localization [1], geoacoustic inversion [2], and tomography [3]. The perturbation-based monitoring of the SSP is the main means of acoustically monitoring the ocean. Dynamic activities in the ocean, ranging from the global climate to internal waves and turbulence, can be observed and analyzed by inverting the SSP [4–6].

Because profile measurements are time-consuming and laborious, it is almost impossible to obtain the real-time three-dimensional (3D) structure of the SSP over a large scale by means of in situ measurements. Researchers have identified a close link between surface parameters and subsurface profiles, and the remote sensing platform with a high spatial and temporal resolution has thus become an important means of obtaining large-scale SSP measurements. Attempts to infer the subsurface profile from satellite observations fall into two categories: "physical" and "statistical" methods. Physical methods take advantage of physical equations between the surface and the subsurface to infer the profile from surface observations. Carnes first discussed the relationship between the sea level (SL) and amplitudes of the empirical orthogonal function (EOF) of the temperature profile [7]. Profile estimation based on the sea surface temperature (SST) and the SL was subsequently found to offer considerable improvements over that based only on the SL, and temperature profiles of the northwest Pacific and northwest Atlantic Oceans were estimated using a single empirical orthogonal function-based regression (sEOF-r) [8]. The sEOF-r method uses an approximate linear equation to describe the physical relationship between parameters of the surface and the subsurface and has been used in such prediction schemes as the operational marine environment of the United States Navy and the Modular Ocean Data Assimilation System (MODAS) [9,10]. Chen confirmed that the sEOF-r method can also be used to directly estimate the global SSP [11]. Although the approximate linear physical relationship inevitably incurs errors owing to the highly nonlinear dynamics of the ocean, research has shown that simple physical expressions can be used to obtain results with reasonably high accuracy. The SST and SL are also effective predictors of the sea surface in the context of profile estimation [12]. Based on big data theory, "statistical" methods describe the relationship between the parameters of the surface and the subsurface through machine learning without preset equations of a specific form. Hjelmervik used the clustering of EOFs and gradient search to estimate the real-time temperature and salinity of the ocean [13,14]. By using the AVISO satellite product as a predictor of the input, Chapman reconstructed velocities by searching for the best-matching class through self-organizing maps (SOM) [15,16]. Su proposed an ensemble learning algorithm that combines extreme gradient boosting and the gradient boosting decision tree to retrieve the profiles of temperature and salinity in the upper 2000 m of the global ocean [17]. Statistical methods can improve the accuracy of profile estimation to a greater extent than physical methods by eliminating the constraints on equation-based inversion.

However, the SST and SL do not include details on the state of the ocean, and this reduces the accuracy of the SSP profile estimated by using them. Based on observations from the central Arabian Sea, Jain found that most errors in reconstructing the SSP occurred at depths of 40–125 m owing to insufficient information about the depth of the mixed layer provided by the SST and SL [18]. Similar errors occurred in SSP estimation in the South China Sea, where the SSP inferred from the SST and SL incurred large errors at a depth close to 500 m due to the exchange of water between the South China Sea and the Pacific Ocean in the Luzhou Strait. This could not be described by remote sensing parameters alone, so multisource parameters were added to strengthen the study of this area [19]. Huang conducted a visual analysis of the practicality of EOF in the South China Sea [20]. Considering that the information available for estimating the subsurface profile is limited, additional predictors are needed to offer richer information to improve the estimation of the SSP. Bao obtained the sea surface salinity (SSS) from both in situ and satellite observations to improve the results of the salinity profile reconstruction [21]. Ou and Chapman estimated SSP using a machine learning method based on SST and SL data [22]. Chen included inverted data from the echo sounder and the depth of the mixed layer in addition to the SST and SL to estimate the SSP [23]. The results indicated that multisource observations can significantly improve the results of estimation, and different predictors make varying contributions to the improvement in estimation.

With advances in the technologies used to observe the global ocean surface, such as voluntary observing ships and moored platforms, the spatiotemporal coverage of in situ surface measurements has significantly increased. These data may help avoid time-consuming and laborious profile measurements. In this study, we improve estimations of the SSP by combining satellite data with in situ observations of the sea surface. We propose a technique for SSP estimation that uses the SOM with multisource observations to this end. We infer the nonlinear relationships between the multisource surface information and the amplitude of the EOF of the SSP by taking advantage of the topology of a neural network cell. The proposed method can be used for the fused processing of multiple parameters from different types of sensors and provides a tool to evaluate the effects of different predictors of the inversion model. Different models are examined to determine the contributions of different in situ surface parameters. The in situ surface parameters that can provide the value of the gain owing to the inversion scheme are identified and an optimized multi-source model is provided. The results of the SSP estimation based on data from the Kuroshio Extension Observatory (KEO) show that the proposed method can improve the accuracy of the acquired SSP. The multisource method has significant advantages in dynamic areas of the ocean, such as the Kuroshio Extension region. In the application of real-time and large-scale subsurface profile estimation, the accuracy may be greatly improved by a surface-going vessel, an automated glider, a mooring station or other in situ surface observation equipment.

## 2. Methodology

The SOM is used to process multisource information for an inversion problem in order to estimate the SSP by combining satellite data with in situ sea surface observations. The processing flow of the SSP estimation is shown in Figure 1.

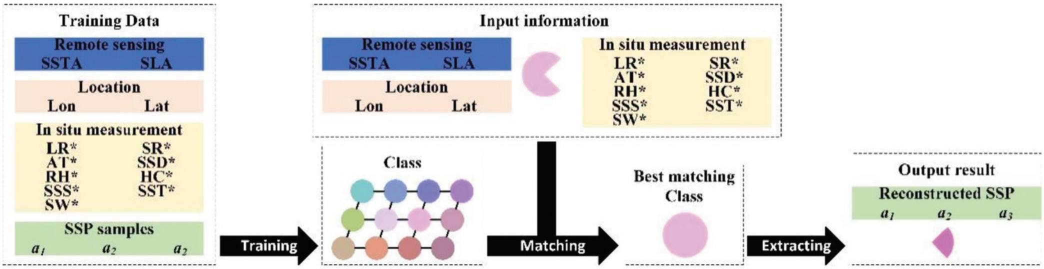


**Figure 1.** The flow of SOM-based estimation of the SSP.

Using SST and SL parameters, previous studies have focused on estimating the SSP. Despite the use of excellent machine learning algorithms, it is still difficult to solve the problem of insufficient information in SST and SL. With the development of measurement technology, more and more parameters can be observed. However, the processing method of parameters is not synchronized. It is not possible to process additional parameters and it is not clear how the parameters affect the results.

The EOF is commonly used in SSP modeling to provide a constraint on the inversion problem. The SPP $c(z,t)$, at a sampling depth $z$, and time $t$ can be described by [24]

$$c(z,t) = c_0(z) + \sum_{s=1}^{\infty} a_s(t) K_s(z) \tag{1}$$

where $c_0(z)$ is the background profile, $K$ is the EOF, and $a$ is the projection coefficient of the EOF. The background profile is the constant part of the SSP that is stable in the long term and can be approximated by the profile of the climatological mean. The superpositions are parts representing perturbations in the SSP. As higher orders of $s$ often introduce excessive noise to the samples, the superposition series are commonly truncated without risking a loss of information from the SSP. A threshold of 95%, which is the proportion of variances, is commonly used to determine the number of modes of the EOF used. According to an analysis of the experimental data, three orders of EOFs are used in SSP modeling here.

The EOF can be calculated from a principal component analysis of the space–time samples. The matrix of anomalies in the SSP of the ocean is $X = [x_1, \cdots x_M] \in \mathbb{R}^{Z \times S}$. It is obtained by sampling over $Z$ discrete points in terms of depth and $S$ instants in time, and by subtracting the background profile from the $S$ samples. Based on singular value decomposition, the EOFs can be calculated by

$$XX^T = K\Lambda^2 K^T \tag{2}$$

where the non-zero elements of $\Lambda^2 = diag([\lambda_1^2, \cdots \lambda_{\rm n}^2]) \in \mathbb{R}^{L \times L}$ represent the variance along the principal directions defined by the corresponding EOF. The three EOFs with the highest variances are used to reconstruct the SSP. In SSP inversion, the EOF vectors can be obtained by the principal component analysis of the samples. Both the input and the output SSPs are expressed in the form of projection coefficients.

A SOM-based estimation technology is proposed to process multisource information for the SSP. The SOM is a nonlinear vector projection algorithm between the input and output layers. In the input layer, each set of measurement data forms a prototype vector, including remote sensing measurements, location data, in situ measurements, and information on samples of the SSP. To evaluate the contributions of different in situ surface parameters, the in situ measurement parameters are considered optional. In the training process, the prototype vectors follow the probability density of the input layer without changing the topological structure. Reference vectors are assigned through the iterative learning algorithm based on the mean values of different classes of all clusterings of the training data. In the output layer, each neuron unit is represented by a class that contains one reference vector. The input information can be regarded as a fragmentary neuron unit, and the missing part represents the coefficients of EOF of the SSP to be estimated. After training the SOM, the input information is matched with different classes on it according to the Euclidian distance. Charantonis introduced a formula to calculate the Euclidian distance over only dimensions with the available parameters [16]:

$$D_E^p(X, Y^p) = \sum_{i \in a} (1 + \sum_{j \in b} (C_{i,j}^p)^2) \times (X_i - Y_i^p)^2 \tag{3}$$

where $D_E$ is the Euclidian distance, $X$ is the input vector, $Y$ is the reference vector, $p$ is the index of each class, $a$ is the set of input data (available variables), and $b$ is the reconstructed output (missing variables) to be solved for. $C$ is the correlation matrix between the missing and the available variables. The input information can be regarded as a neuron with missing information, representing the projection coefficients that describe the SSP. The reference vector closest to the input vector can be identified from Equation (3) by using the best-matching class. The missing projection coefficients of the input vector can then be estimated by extracting the corresponding part of the best-matching class and outputting it to reconstruct the SSP.

## 3. Data

We used surface data collected from in situ measurements and satellite observations as input parameters for the proposed model. Part of the subsurface measurements of the sound speed profile was used to train the SOM and the rest to validate the accuracy of the estimated SSP. The climatological mean profile was used as the background profile.

All in situ surface measurements and SSP samples used were from the KEO. As a surface mooring, the KEO (32.3° N, 144.6° E) has a long record of daily real-time measurements in the ocean. The slack-line mooring provides a rich variety of surface and subsurface data, including longwave radiation (LR), shortwave radiation (SR), wind speed (WS), surface temperature (ST), surface salinity (SS), air temperature (AT), surface density (SD), heat content (HC), relative humidity (RH), temperature profile, and salinity profile. The temperature and salinity profiles of the SSP samples can be calculated from Del Grosso's empirical formula [25]. As EOF processing requires samples at the same depth, the SSP sample is considered completed only when its sample depths are shallower than 5 m and deeper than 475 m. And the remaining profiles were cubic interpolated to the nominal depths of sensors on the slack-line (5, 10, 15, 20, 25, 35, 40, 50, 75, 100, 125, 150, 175, 200, 225, 275, 325, 400, 425, 475). For the KEO, such a depth range can retain more perturbation description while losing too many samples due to the exclusion of shallower profiles. A total of 2277 profiles were finally obtained. Of them, 80% (1820 samples from 26 September 2009 to 3 October 2018) formed the training dataset, and the other 20% (457 samples from 4 October 2018 to 5 May 2020) were used as the test set.

The WOA18 was used to describe the background profile. It was obtained from the National Oceanic and Atmospheric Administration. It analyzes in situ measurements from a wide variety of sources and provides the global gridded mean climatological profile. The annual average profile from 2005 to 2017 was used at a spatial resolution of 0.25°. Figure 2 shows all the SSPs used from the experimental area. Owing to the intense exchange of energy and matter in the Kuroshio Extension, a large perturbation of about 30 m/s in the sound speed occurred on the sea surface and at a depth of 475 m. The large amplitude of the perturbation, with a complex origin, posed a challenge to the SSP estimation.

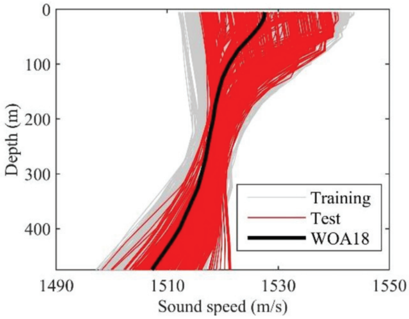


**Figure 2.** Profiles of training, testing, and the background.

The surface parameters of satellite remoting sensing used here were collected from the Coriolis projects (https://marine.copernicus.eu (accessed on 29 January 2021)). So that they could be distinguished from the surface parameters of the in situ measurements, the remoting sensing parameters consisted of anomalies in the sea level (SLA) and the sea surface temperature (SSTA), and had a spatial resolution of 0.25° and a temporal resolution of one day.

## 4. Results

Eleven models of the surface parameters derived from the satellite observations and in situ measurements were evaluated, and the results are listed in Table 1 and shown in Figure 3. The precision of the SSP reconstruction was difficult to ensure owing to the complex and intense perturbance in the SSP in the Kuroshio Extension. The mean-variance of the SSP samples was 5.90 m/s, with the appearance of a perturbance of large amplitude in both the sea surface and the thermocline measurements. Model 1 was the classic model for estimating the SSP, and was based only on remote sensing data. It was used as a reference. Model 2–10 evaluated the effects of different sea surface parameters obtained from in situ measurements. If the error in the models examined was smaller than that of Model 1, the in situ parameters were considered to have provided effective information to estimate the SSP and improve the accuracy of the results. Conversely, if the error in the model was larger than that of Model 1, this indicated that the parameters of the in situ measurements had led to redundant neural network topology and the mirage had reduced the accuracy of the results of inversion. As parameters directly affect the sound speed, temperature, salinity, and density were effective input parameters, and the temperature had the most significant effect on the accuracy of SSP estimation. Although air temperature and heat content do not directly reflect the physical properties of seawater, they can reduce error, and show a strong correlation with the SSP. The heat content was a special parameter that led to the greatest improvement in accuracy at a depth of about 50 m from the sea surface. The addition of longwave radiation, shortwave radiation, and wind speed directly increased the error, and so these parameters were considered inappropriate input parameters for SSP estimation. To obtain the best estimation results, all effective in situ predictors and remote sensing predictors were combined in Model 11, which delivered the best results of all parameter combinations considered.

**Table 1.** Results of SSP estimation of each model.

| Model | Parameters | Root Mean Square Error (m/s) | Mean Absolute Error (m/s) |
|---|---|---|---|
| 1 | SLA, SST | 3.49 | 2.52 |
| 2 | SLA, SST, LR | 3.56 | 2.55 |
| 3 | SLA, SST, SR | 3.79 | 2.56 |
| 4 | SLA, SST, WS | 3.69 | 2.59 |
| 5 | SLA, SST, ST | 2.71 | 1.97 |
| 6 | SLA, SST, SS | 3.07 | 2.25 |
| 7 | SLA, SST, AT | 3.33 | 2.45 |
| 8 | SLA, SST, SD | 2.75 | 2.03 |
| 9 | SLA, SST, HC | 2.82 | 2.03 |
| 10 | SLA, SST, RH | 3.63 | 2.54 |
| 11 | SLA, SST, ST, SS, AT, SD, HC | 2.18 | 1.56 |

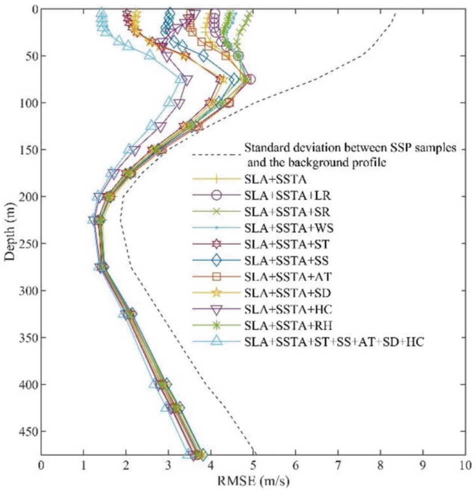


**Figure 3.** Results of SSP estimation for each model.

According to the variation in the estimation error with depth, it is clear that the accuracy of SSP reconstruction was related to the amplitude of perturbations. A large anomaly in the sound speed might have led to a larger error. Because the results were directly derived from the sea surface parameters, accuracy was relatively high near the sea surface even if there was a large deviation in the speed of sound. In the range of depth of 50–150 m below the surface, large errors occurred owing to seasonal and diurnal variations in the mixed layer. The differences between the models gradually decreased with increasing depth, and they delivered nearly the same performance below 250 m. The results of inversion show that the introduction of in situ measurement data improved the accuracy of SSP estimation. This was mainly reflected at depths close to the sea surface. In the case of a large perturbation deeper below the surface, the accuracy did not improve when only the sea surface parameters were considered. To examine the performance of the multisource method in practice, we assessed Model 11 further.

The estimation errors of all samples are shown in Figure 4. Except for a few samples, the multisource model improved the accuracy and robustness of the results. The standard deviation of the estimated SSP was 5.54 m/s and the maximum value was 9.73 m/s. Model 1 had an overall root mean square error (RMSE) of 3.49 with a maximum error of 11.10 m/s. Although the remote sensing parameters were effective predictors as inputs to the inversion model, insufficient information led to many incorrect results. The multisource model performed better than model 1 in about 80% of the samples. Its RMSE and the maximum error were 2.18 m/s and 6.66 m/s, respectively. In addition, about 82% of the estimated SSPs of the multisource model were within the error limit of 3 m/s, and this value decreased to only 58% without the in situ measurements.

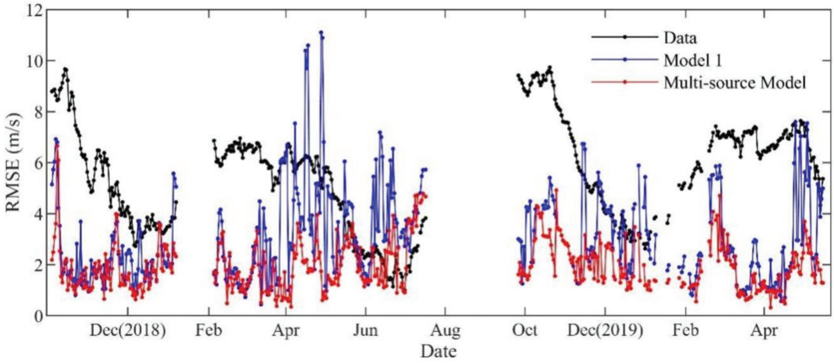


**Figure 4.** Deviations in the data and estimation errors for different samples.

Some parameters might have assisted in the estimation in a special way. Prediction in this case, which might have degraded the results of estimation owing to drastic changes in value, can indicate anomalies in the results of inversion. Almost all failures of estimation corresponded to a large predicted value, which reflected perturbance owing to extreme weather events, such as typhoons and rainstorms. This can be used to build an early warning index for the reliability of the results.

Figure 5 shows the estimated speeds of sound by two models at all depths. Model 1, based only on remote sensing parameters, had an overall mean absolute error of 2.52 m/s, with a value of the coefficient of determination ($R^2$) of 0.76 and a slope of 37.2. The multisource model outperformed model 1, with an absolute error of 1.56 m/s, $R^2$ of 0.90, and slope of 43.9. The speed of sound estimated by it had an error of only 0.1% compared with the measured speed of sound.

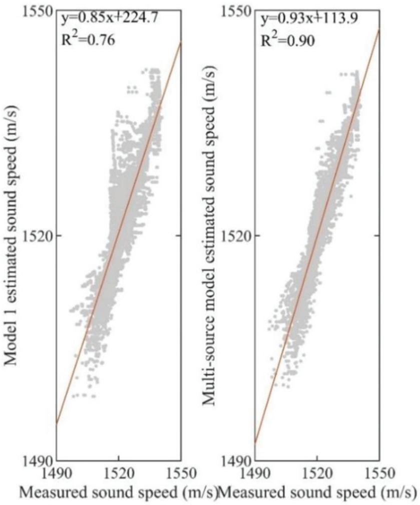


**Figure 5.** Scatter in the measured and estimated speeds of sound obtained by different models.

The SSPs estimated by different models (for the first sample in each month in the period considered) are given in Figure 6. Consistent with the previous statistical results, the results of the multisource model were significantly better than those of the model based only on remote sensing parameters. Large errors mainly appeared in parts with large perturbances in the speed of sound, mainly at depths of less than 150 m and greater than 350 m. In the case of perturbations on the sea surface, the accuracy of the estimated SSP significantly improved after the introduction of multisource information, such as in the results for 4 October 2018 and 1 April 2019. In other regions, the perturbance in the sound speed generally decreased with depth. Most errors occurred in the upper thermocline, possibly due to variations in the mixed layer or the dynamics of the ocean, such as internal waves. The multisource method significantly improved the accuracy of the estimated sound speed in the upper ocean. A special characteristic of the KEO is that due to the exchange of energy and matter, perturbations in the deeper thermocline do not decrease to less than in the surface layer. The results show that it was difficult to estimate perturbations far from the surface because all the input parameters were based on data obtained on the sea surface. Although modes of the EOF can express the SSP at all depths as a whole, the principal components of the shape of perturbance had no significant correlation between the surface and deeper layers of the ocean. In addition, different models obtained consistent results on data from 1 November 2019 and 1 April 2020. This indicates that inversion processing and EOF representation cannot be used by themselves to accurately reconstruct the profile of sound speed.

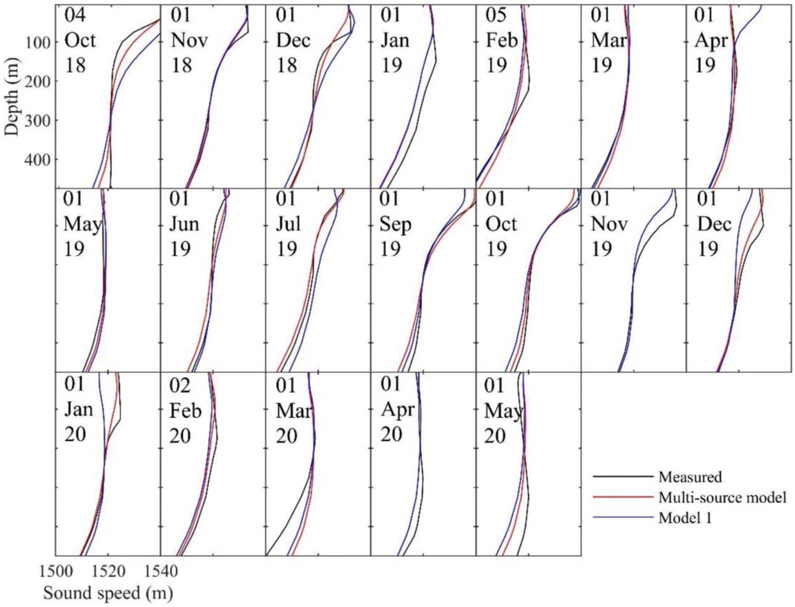


**Figure 6.** Comparison between the measured and the estimated SSPs.

## 5. Conclusions

This study proposed a method to estimate the SSP based on the SOM. We used the topological structure of neurons to input multisource observations and estimate the SSP. We used data from the Kuroshio Extension to assess the performance of the proposed model, which uses different surface parameters derived from remote sensing and in situ measurements. The results showed that the addition of parameters of in situ measurements can markedly improve the estimation of SSP. Compared with model 1, the accuracy of the multisource model is improved by 38%. However, not all surface parameters play a positive role in the inversion. The best results were obtained when remote sensing data and effective predictors were used in the multisource model. Error in the estimated SSP in the region of the Kuroshio Extension was larger than in other parts of the ocean owing to the intense exchange of energy and matter there. The degradation in accuracy revealed that sea surface parameters can be used to accurately estimate the sound speed in the surface layer of the ocean but not in the deep layers. The large amplitude of SSP perturbances poses a challenge to the accurate estimation of the SSP.

Further work is needed to accurately calculate profiles of the SSP as almost all currently available methods are statistical, and require physical rules to obtain reliable estimations. Moreover, methods that can be applied to regions for which few samples are available should be researched.

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

**Funding:** This research was funded by the Natural Science Foundation of Guangdong Province, grant number (No. 2022A1515011519) and the 2022 Basic Research funds for central universities of Northwestern Polytechnical University.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**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.

## References

1. Michalopoulou, Z.H.; Gerstoft, P.; Caviedes-Nozal, D. Matched field source localization with Gaussian processes. *JASA Express Lett.* **2021**, *1*, 064801. [CrossRef]
2. Tan, T.W.; Godin, O.A.; Katsnelson, B.G. Passive geoacoustic inversion in the Mid-Atlantic Bight in the presence of strong water column variability. *J. Acoust. Soc. Am.* **2020**, *147*, EL453–EL459. [CrossRef]
3. Jin, K.; Xu, J.; Wang, Z. Deep learning convolutional neural network applying for the Arctic acoustic tomography current inversion accuracy improvement. *J. Mar. Sci. Eng.* **2021**, *9*, 755. [CrossRef]
4. Zhang, S.; Piao, S. Broadband Sound Intensity Interference Frequency Periodicity and Pulse Source Localization. *J. Mar. Sci. Eng.* **2021**, *9*, 200. [CrossRef]
5. Yang, T.C.; Huang, C.; Huang, S. Frequency striations induced by moving nonlinear internal waves and applications. *IEEE J. Ocean. Eng.* **2017**, *42*, 663–671. [CrossRef]
6. Dushaw, B.D. Acoustic thermometry as a component of the global ocean observing system. *J. Acoust. Soc. Am.* **2012**, *132*, 142–149. [CrossRef]
7. Carnes, M.R.; Mitchell, J.L.; Witt, P.W. Synthetic temperature profiles derived from Geosat altimetry: Comparison with air-dropped expendable bathythermograph profiles. *J. Geophys. Res. Oceans* **1990**, *95*, 17979–17992. [CrossRef]
8. Carnes, M.R.; Teague, W.J.; Mitchell, J.L. Inference of subsurface thermohaline structure from fields measurable by satellite. *J. Atmos. Ocean. Technol.* **1994**, *11*, 551–566. [CrossRef]
9. Meijers, A.J.S.; Bindoff, N.L.; Rintoul, S.R. Estimating the four-dimensional structure of the Southern Ocean using satellite altimetry. *J. Atmos. Ocean. Technol.* **2011**, *28*, 548–568. [CrossRef]
10. Rahaman, H.; Behringer, D.W.; Penny, S.G.; Ravichandran, M. Impact of an upgraded model in the NCEP Global Ocean Data Assimilation System: The tropical Indian Ocean. *J. Geophys. Res. Oceans* **2016**, *121*, 8039–8062. [CrossRef]
11. Chen, C.; Ma, Y.; Liu, Y. Reconstructing sound speed profiles worldwide with Sea surface data. *Appl. Ocean Res.* **2018**, *77*, 26–33. [CrossRef]
12. Swart, S.; Speich, S.; Ansorge, I.J. An altimetry-based gravest empirical mode south of Africa: 1. Development and validation. *J. Geophys. Res.* **2010**, *115*, C03002. [CrossRef]
13. Hjelmervik, K.T.; Hjelmervik, K. Estimating temperature and salinity profiles using empirical orthogonal functions and clustering on historical measurements. *Ocean Dyn.* **2013**, *63*, 809–821. [CrossRef]
14. Hjelmervik, K.; Hjelmervik, K.T. Time-calibrated estimates of oceanographic profiles using empirical orthogonal functions and clustering. *Ocean Dyn.* **2014**, *64*, 655–665. [CrossRef]

15. Charantonis, A.A.; Testor, P.; Mortie, L. Completion of a sparse GLIDER database using multi-iterative self-organizing maps (ITCOMP SOM). *Proc. Comput. Sci.* **2015**, *51*, 2198–2206. [CrossRef]
16. Chapman, C.; Charantonis, A.A. Reconstruction of subsurface velocities from satellite observations using iterative self-organizing maps. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 617–620. [CrossRef]
17. Su, H.; Yang, X.; Lu, W. Estimating subsurface thermohaline structure of the global ocean using surface remote sensing observations. *Remote Sens.* **2019**, *11*, 1598. [CrossRef]
18. Jain, S.; Ali, M.M. Estimation of sound speed profiles using artificial neural networks. *IEEE Geosci. Remote Sens. Lett.* **2006**, *3*, 467–470. [CrossRef]
19. Li, H.; Qu, K.; Zhou, J. Reconstructing Sound Sped Profile from Remote Sensing Data: Nonlinear Inversion Based on Self-Organizing Map. *IEEE Access* **2021**, *9*, 109754–109762. [CrossRef]
20. Huang, J.; Luo, Y.; Li, Y.; Shi, J.; Zheng, X.; Wang, J. Analysis of Sound Speed Profile in the South China Sea based on Empirical Orthogonal Function Algorithm. In Proceedings of the 2021 OES China Ocean Acoustics (COA), Harbin, China, 4–17 July 2021; pp. 166–171. [CrossRef]
21. Bao, S.; Zhang, R.; Wang, H. Salinity profile estimation in the Pacific Ocean from satellite surface salinity observations. *J. Atmos. Ocean. Technol.* **2019**, *36*, 53–68. [CrossRef]
22. Ou, Z.; Qu, K.; Liu, C. Estimation of Sound Speed Profiles Using a Random Forest Model with Satellite Surface Observations. *Shock Vib.* **2022**, *2022*, 2653791. [CrossRef]
23. Cheng, C.; Yan, F.; Gao, Y. Improving reconstruction of sound speed profiles using a self-organizing map method with multi-source observations. *Remote Sens. Lett.* **2020**, *11*, 572–580. [CrossRef]
24. Cheng, L.; Ji, X.; Zhao, H. Tensor-based basis function learning for three-dimensional sound speed fields. *J. Acoust. Soc. Am.* **2022**, *151*, 269–285. [CrossRef]
25. Grosso, V.A.D. New equation for the speed of sound in natural waters (with comparisons to other equations). *J. Acoust. Soc. Am.* **1974**, *56*, 1084–1091. [CrossRef]

*electronics*

*Article*

# Transformer-Based Distillation Hash Learning for Image Retrieval

**Yuanhai Lv [1,2], Chongyan Wang [3], Wanteng Yuan [4], Xiaohao Qian [1], Wujun Yang [2,*] and Wanqing Zhao [1]**

[1] School of Information Science and Technology, Northwestern University, Xi'an 710127, China
[2] Information Network Center, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[3] Social Cooperation Department, Xi'an University of Finance and Economics, Xi'an 710100, China
[4] Network Technology Department, Xi'an Aeronautics Computing Technique Research Institute, Xi'an 710065, China
* Correspondence: wujun@xupt.edu.cn

**Abstract:** In recent years, Transformer has become a very popular architecture in deep learning and has also achieved the same state-of-the-art performance as convolutional neural networks on multiple image recognition baselines. Transformer can obtain global perceptual fields through a self-attention mechanism and can enhance the weights of unique discriminable features for image retrieval tasks to improve the retrieval quality. However, Transformer is computationally intensive and finds it difficult to satisfy real-time requirements when used for retrieval tasks. In this paper, we propose a Transformer-based image hash learning framework and compress the constructed framework to perform efficient image retrieval using knowledge distillation. By combining the self-attention mechanism of the Transformer model, the image hash code is enabled to be global and unique. At the same time, this advantage is instilled into the efficient lightweight model by knowledge distillation, thus reducing the computational complexity and having the advantage of an attention mechanism in the Transformer. The experimental results on the MIRFlickr-25K dataset and NUS-WIDE dataset show that our approach can effectively improve the accuracy and efficiency of image retrieval.

**Keywords:** image retrieval; Transformer; self-attention; knowledge distillation; hashing learning

**Citation:** Lv, Y.; Wang, C.; Yuan, W.; Qian, X.; Yang, W.; Zhao, W. Transformer-Based Distillation Hash Learning for Image Retrieval. *Electronics* **2022**, *11*, 2810. https://doi.org/10.3390/electronics11182810

Academic Editor: Yue Wu

Received: 18 August 2022
Accepted: 4 September 2022
Published: 6 September 2022

## 1. Introduction

With the development of WEB 2.0, Internet information generation has changed from traditional website employee generation to user-led generation. Virtual social activities with the help of various multimedia social platforms are becoming increasingly normalized, and the human-oriented self-media have led to a qualitative change in the status and role of people in Internet activities. From the traditional information receiver to the information publisher, more and more people are communicating by publishing and sharing multimedia information. Birjandi et al. proposed the use of text to retrieve image content (KBIR, keyword-based image retrieval) to meet the needs of users for large-scale image retrieval [1]. By manual annotation, the method uses keyword attributes between text and images to build an index. However, manual annotation is labor-intensive in many cases. Users are often unable to describe the content of an image, so researchers have proposed a technique that allows users to enter images to search for relevant images (content-based image retrieval—CBIR) [2]. These methods have automatically extracted the image content features by analyzing the image content and then quantifying them, and building indexes for retrieval based on the quantified content features. However, because traditional content-based image retrieval methods usually use manual features, which is a fixed representation of visual features, and lack the ability to learn, retrieval performance is challenging to improve. With the study of deep convolutional neural networks and the large-scale accumulation of image data, many methods are using deep convolutional neural networks (DCNNs) to automatically learn the features of images and use these features to retrieve images from large-scale datasets. Peng et al. proposed an image retrieval method based on DCNNs and binary hash learning [3]. The method uses DCNNs to learn the intrinsic distribution of images and extract image features while adding a hashing layer to the DCNNs to learn deep features and hash codes to perform an effective retrieval.

In recent years, the Transformer has become a prevalent architecture in the field of deep learning. For example, BERT [4] and GPT-2 [5], which have become very famous works in the natural language processing (NLP) field [6] in recent years, use the Transformer architecture. Transformer relies on a simple but potent mechanism, the Attention mechanism [7], which allows neural network models to selectively focus on certain parts of the input and reason more efficiently. Various architectures based on Transformer have been used with encouraging results in sequence prediction, language modeling, and machine translation.

Convolutional computation has better locality and spatial invariance and has an inherent natural advantage (inductive bias) for visual problems. The CNN model [8] needs to obtain a larger perceptual field by continuously stacking Conv layers [9]. Moreover, the number of operations required to compute the association between two locations increases with the location distance. The self-attention is the fundamental component of Transformer [7], and the number of operations required by self-attention to compute the association between two positions is distance independent. The inherent advantage is that, unlike convolution which has a fixed and limited field of perception, the self-attention operation can obtain long-range information [10]. By examining the weight distribution of each Attention Head, self-attention can produce more interpretable results.

Nowadays, inspired by the Transformer in NLP, researchers have extended the Transformer to the field of computer vision (CV). Dosovitskiy et al. proposed ViT [11], a model that feeds a sequence of image blocks (Patches) into a standard Transformer. This framework achieves the same state-of-the-art results on multiple computer vision task baselines. Some Transformer architectures for image retrieval tasks have also been gradually proposed [12]. These apply Transformer to pre-trained models for feature extraction, and then perform a similarity comparison in the feature space to solve problems in image retrieval.

Transformer can obtain the global sense field by self-attention mechanism, and can enhance the weight of unique discriminable features for image retrieval tasks to improve the retrieval quality. However, Transformer is computationally intensive, and it is challenging for it to meet real-time requirements when used for retrieval tasks. Knowledge distillation can replace a teacher model with a lightweight student model to improve speed and ensure accuracy. Therefore, it is necessary to design a framework based on knowledge distillation [13] to improve the retrieval speed for the Transformer-based image retrieval model. In recent years, the parameter size of models has become larger and larger, which often requires a large number of memory resources and is time-consuming to run during the deployment phase of the model. By model distillation, the model with huge parameters is compressed into a small parametric number model, which can make the model occupy fewer resources and be less time-consuming without losing accuracy.

This paper proposes a Transformer-based image hash learning framework, and the constructed framework is compressed for efficient image retrieval using knowledge distillation. First, Resnet-based Backbone is used to extract the image features. Then, the features are input to multi-head of Transformer as query, key, and value, respectively, and the mutual attention weights are calculated using Transformer. Then, the features fused by Transformer are mapped to Hamming space to perform more compact hash learning. Finally, the performance of this model is distilled into a smaller and faster student model for real-time retrieval.

The main contributions of the proposed method include: (1) by introducing the self-attention mechanism of the Transformer model, it makes the image features a global perspective, which can strengthen the weights of unique discernible features and improve the image retrieval quality. (2) By introducing knowledge distillation, the computationally heavy Transformer model is compressed into an efficient lightweight model, which reduces the computation and also has the advantage of features learned by the Transformer. (3) The experimental results on two public datasets achieved encouraging results, demonstrating the effectiveness of the proposed method.

## 2. Related Works

In recent years, many machine learning-based search methods have been proposed to perform an efficient search of multimedia data. Among them, hash-based methods are gradually becoming the majority approaches for this problem [14–19]. These methods mainly use a variety of hash functions to encode the high-dimensional feature of an image to a low-dimensional representation, while expecting to preserve the approximate relationships in the original space after mapping them. Traditional hashing methods usually use manual image features such as SIFT [20], HOG [21], etc., to extract image features and then transform the features into hash codes using a fixed hash mapping function. This will reduce the capacity to express visual content and cannot handle complex similarity semantics well. Chugh et al. [22] combined multiple features to improve the retrieval of plants. With the popularity of deep convolutional neural networks (CNNs), CNNs are also gradually used for hash learning to solve the above problems. CNN-based supervised hash learning methods have achieved groundbreaking experimental results on many baselines [17–19,23–25]. For example, Xia et al. [23] proposed a CNN-based hashing method that learns binary hash codes by supervised training and demonstrates significant search performance on some public baselines. Zhang et al. [24] proposed to increase the rule element of the loss function based on triplet learning for the supervised deep hash coding method by using a Laplacian matrix. Furthermore, the method achieves bit scalability by giving a weight to each bit of the hash coding. The approach proposed by Zhao et al. [19] learns the hash coding of objects, while the method implements weakly supervised hash coding using multi-instance learning.

Google first proposed the Transformer model in 2017, which was first proposed by Vaswani et al. [26], as an only attention-based mechanism to implement machine translation tasks. Subsequently, Devlin et al. [4] pre-trained the Transformer by letting the model predict the masked words on untagged text data. This approach (i.e., BERT) was later considered a new paradigm for natural language representation models. Inspired by the Transformer in NLP, researchers have extended this mechanism to the field of computer vision (CV). In contrast to the previous CNN models [27], Chen et al. [28] pre-trained a sequence Transformer to predict masked pixels and was much more effective than CNN on image classification tasks. Dosovitskiy et al. proposed ViT [11], which applies the standard Transformer to image block (Patch) sequences for learning the embedded representation sequences of blocks. The output of the Transformer encoder is used as a representation and prediction of the image, which makes it equally state-of-the-art performance on multiple image recognition baselines. Transformer also achieved the desired results in high-level vision (HLV) tasks. High-level vision tasks are concerned with understanding and using the semantic content in images [29]. DETR [30] attempts to solve the image object detection problem using the Transformer, which treats the object detection task as an image-to-set prediction problem and simplifies detecting images. To address the limitations of the Transformer Attention module in processing image feature maps, researchers have further proposed deformable DETR for end-to-end object detection [31]. Since DETR still heavily relies on object box prediction during the training process, Wang et al. [32] proposed a dual-path Transformer for end-to-end panoramic segmentation, which effectively unifies semantic segmentation and instance segmentation.

Transformer has also gained some attention in the field of image retrieval. Liu et al. proposed the first large-scale text-to-image retrieval (VisualSparta) based on Transformer [33]. The proposed method can retrieve relevant images from a large and unlabeled set of images under a given text query. Facebook proposed a visual Transformer-based image retrieval model [12]. The model uses a visual Transformer to generate image descriptors and trains the model with a metric learning objective. The metric learning objective combines contrast loss with a differential entropy regularizer. In the Google Landmark Recognition 2021 Kaggle competition, Henkel et al. [34] proposed an effective end-to-end network architecture for large-scale landmark recognition and retrieval. The network architecture combines DLOG (orthogonal Local and Global) [35] and the hybrid Swin–Transformer model [36] and uses a predictive retrieval approach. For each query image, the method uses L2 to normalize the cosine similarity between image descriptors and then searches for the most similar image in the indexed image database.

Although many Transformer-based models have been proposed to solve computer vision tasks such as image retrieval, existing Transformer models are usually huge and computationally expensive, e.g., the base ViT model [11] requires 18 billion FLOPs to process the images. The knowledge distillation algorithm [13] is a technique that can compress a large network into a small efficient network and obtain comparable performance. The knowledge distillation algorithm usually uses a teacher network (large model) to teach a student network (small model), transferring the knowledge from the teacher network to the student network so that the performance of the student model is as close as possible to the performance of the teacher model. In 2015, Hinton et al. [13] first proposed the concept of knowledge distillation (KD) in neural networks, using the output logits of the teacher network or the integrated network (many teacher networks) and applying these logits to train fast small networks. Remero et al. [37] used not only the final output logits of the teacher network, but also its intermediate hidden layer parameter values (intermediate representations) to train the student network. Mirzadeh et al. [38] introduced multi-step knowledge distillation in teacher networks and student networks with teaching assistant networks. Knowledge distillation can be used for many networks (e.g., from large to small networks, from single networks to integrated networks, from CNNs to Transformer, etc.). More relevantly, Tian et al. [39] used distillation learning to fuse multi-modal transformers for a sketch-based image retrieval task. Few studies are currently related to knowledge distillation on Transformer-based image hashing learning networks. Our work focuses on using Transformer to obtain hash codes with a global view and high uniqueness and using knowledge distillation to obtain lightweight and efficient models that can be deployed in practice.

## 3. Method

This section describes our proposed framework, including a Transformer model for learning hash representations with a global view and high uniqueness and a lightweight and efficient model obtained by distillation that can be used for practical deployment. Section 3.1 introduces the architecture of the proposed framework in this work. Section 3.2 describes a CNN-based backbone for image feature extraction. Section 3.3 describes a Transformer teacher module for high-level semantic hash learning and a lightweight and efficient student model. Section 3.1 introduces the training phase including a ranking loss based on triplet samples and a distillation learning loss.

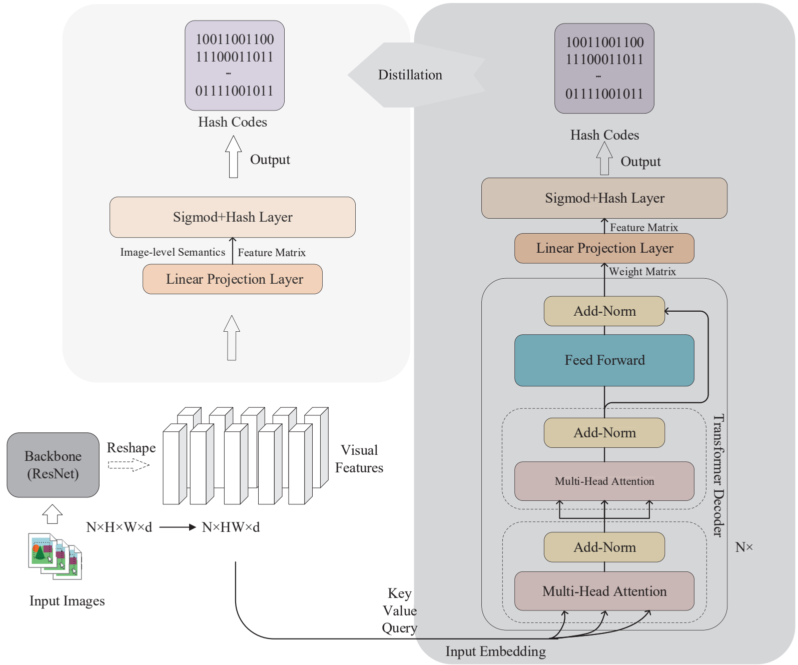
### *3.1. Model Overview*

This work first extracts the visual features of the images using a CNN-based backbone. Then, the decoder of Transformer is used to perform the fusion of different image patch features by self-attention. Moreover, the final fused features are mapped to Hamming space to perform more compact hash learning. Finally, the knowledge learned by the Transformer module is distilled into a smaller and faster model.

In this paper, backbone utilizes a pre-trained ResNet model [27]. The backbone model is mainly used for feature extraction on images, so that this neural network model is generic and can be integrated with any advanced deep model. These methods can also be trained on unlabeled or weakly labeled data, thus further improving their performance.

The cross-attention described in this work utilizes the decoder in the Transformer structure to perform cross-attention. In this method, the query, key, and value in the cross-attention layer in the standard decoder are all from the image features output by the backbone, and the process of computing attention by all three completes the cross-attention among visual features and achieves feature fusion.

In hash learning, the fused features are mapped into compact pseudo-binary codes that are used to improve the efficiency of image retrieval. The same image features from the backbone are input in the student module. However, instead of performing Transformer's attention computation, the linear layer is directly used to perform accelerated projection and constrain the output to be consistent with the Transformer. Figure 1 illustrates the overall architecture of our proposed framework.



* Residual connections are used around the three sub-layers of each layer in the Transformer architecture, and layer normalization is performed.

**Figure 1.** The overall architecture of our proposed framework.

### *3.2. CNN-Based Backbone*

In our work, the main role of the backbone is to perform the initial feature extraction of the image. Our backbone uses the ResNet-50 model [27]. Figure 2 is a brief framework diagram of ResNet-50. In ResNet-50, residual learning is applied to every few stacked layers to construct a residual block, defined as:

$$\mathbf{y} = F(\mathbf{x}, \mathbf{W}) + \mathbf{x}, \tag{1}$$

where $\mathbf{x}, \mathbf{y}$ are the input and output vectors for computing the current residual block. $F(\mathbf{x}, \mathbf{W})$ denotes the residual mapping function, and $\mathbf{W}$ is the parameters to be learned, $F(\mathbf{x}, \mathbf{W}) + \mathbf{x}$ is achieved by adding the shortcut linking layers and elements, and the dimensions must be equal before their summation.

In our work, we remove the final averaging pooling layer and full connection layer of Resnet-50, and use convolutional computation to extract its spatial features $\mathcal{F}_0 = \mathbb{R}^{H \times W \times d}$, where $H \times W$ represent the height and weight of the feature map, respectively, and $d$ denotes the dimension of features. We set $H = W = 18$, so in total there are 324 elements of feature embedding vectors. Since the output dimension of ResNet-50 is 2048, we set the dimension of our embedding size $d$ as 2048.

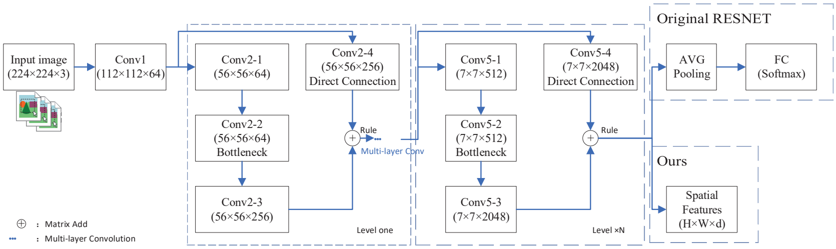


**Figure 2.** A brief framework diagram of ResNet-50.

### 3.3. Transformer Teacher Module and Student Module

**Query update:** The visual features $\mathcal{F}_0$ of the image obtained from the above backbone are then input into the decoder of Transformer as key, value, and query. The image's spatial features are calculated using a multi-layer Transformer for self-attention. The input image is divided into $d$ embedding vectors, and each embedding vector obtains a one-dimensional position encoding token corresponding to its position and is used as input to the decoder regularly. The prior position is merged by adding a learnable one-dimensional position encoding code to the input embedding vectors. An extra learnable CLS token is added to the input embedding vector to represent its correspondence to the output token as a global concept. The Transformer consists of $L$ layers, each layer consisting of two main modules: a multi-headed self-attentive (MSA) layer, which applies the self-attentive operation to different embedding vectors of the inputs, and a feed-forward network (FFN). Both the MSA and FFN layers are preceded by a normalization layer, and followed by a skip connection layer. The query of the $(i)$-th decoder layer $Q_i$ is updated based on the output of its previous layer $Q_{i-1}$ as follows:

$$Self - Attention : Q_i^{'} = Q_{i-1}^{'} + MultiHead(\widetilde{Q}_{i-1}, \widetilde{Q}_{i-1}, Q_{i-1}) \tag{2}$$

$$FFN : Q_i = FFN(Q_i^{'}) \tag{3}$$

where $\widetilde{Q}_{i-1}$ denotes the embedding vectors with the position encoding of outputs at layer $(i-1)$-th, $Q_i^{'}$ is the intermediate variable, $MultiHead(query, key, value)$ and $FNN(x)$ are the multi-headed attention mechanism and feed-forward network, respectively.

**Pooling:** We need to extract a compact code that globally describes the image. In our reference pooling approach, we directly treat the output of the CLS embedding as a global image descriptor.

**Reduction and binarization:** After obtaining the global image descriptors, we further perform dimensionality reduction and binarization to improve the retrieval speed. Specifically, we use a combination of a linear projection layer and a Sigmoid function to map the global image descriptors through the fully connected layer to a smaller size $b$ bit codes and project the logit values to $[0, 1]$ using the Sigmoid function:

$$\hat{\mathbf{h}} = Sigmoid(\mathbf{W}_k^T \mathbf{z}_{cls} + \mathbf{b}_k) \tag{4}$$

where $\mathbf{W}_k^T$ and $\mathbf{b}_k$ are the parameters of the linear projection layer and $\hat{\mathbf{h}}$ is the pseudo hash code. It is worth mentioning that since the symbolic function sgn() is not derivable, the pseudo hash codes generated in the training phase are real-valued, while in the inference phase, the binary hash codes are generated by the following formula:

$$\mathbf{h} = \frac{1}{2}(sgn(\hat{\mathbf{h}} - 0.5) + 1) \tag{5}$$

where **h** denotes the binary hash code.

**Student module:** The designed student model is straightforward. We directly remove the Transformer part of the teacher model. The feature map generated from the backbone will go through two fully-connected layers and output the same size as the teacher model.

### *3.4. Training*

**Loss function for teacher model:** there are two losses used to train the teacher model, namely metric ranking loss based on triplet samples and quantified loss. The metric ranking loss function is used to make the similarity between positive sample pairs greater than the similarity between negative sample pairs:

$$L_{Triplet} = \sum_{n}[S(I_n, I_n^+) - S(I_n, I_n^-) + \delta]_+ \tag{6}$$

where $[x]_+ = max(0, x)$, $S()$ is the cosine similarity between sample pairs, $(I_n, I_n^+)$ are positive sample pairs and $(I_n, I_n^-)$ are negative sample pairs. The metric ranking loss would treat the sample pair of $S(I_n, I_n^-) + \delta > S(I_n, I_n^+)$ as a valid sample pair and increase the penalty, while for the sample pair of $S(I_n, I_n^-) + \delta \leq S(I_n, I_n^+)$ is considered to satisfy the desired goal and is therefore ignored. By training with this loss, the cosine identity of the negative sample pair will be guaranteed to be at least $\delta$ greater than the cosine identity of the positive sample pair.

The quantization loss in our approach is mainly to constrain the network output pseudo hash codes where the code values are as close to 0 or 1 as possible, as follows, which penalizes the network if the output of a neuron is close to 0.5:

$$L_{Quan} = -\sum_{n}\frac{1}{b}(\hat{\mathbf{h}}_n - 0.5\mathbf{1})^T \cdot (\hat{\mathbf{h}}_n - 0.5\mathbf{1}) \tag{7}$$

where **1** represents a vector of ones of length $b$. During training for the teacher model, we weight the two loss components $L_{Triplet}$ and $L_{Quan}$ by factors $\lambda_1$ and $\lambda_2$, respectively. Therefore, the overall loss for the teacher model is: $L_{Teacher} = \lambda_1 L_{Triplet} + \lambda_2 L_{Quan}$.

**Loss function for student model.** The student model uses the same backbone as the teacher model and both sides share the same backbone parameters. We fix the backbone parameters when training the student model. The subsequent structure is mapped to the same $b$ bits as the output of the teacher model by a fully connected mapping layer and activated with a sigmoid function. Relative entropy is used here as a distillation loss function to measure the distance between the two model distributions. Assuming that, for any sample $n$, the hash code output by the teacher model is $\hat{\mathbf{h}}_n^{tech}$ and the hash code output by the student model is $\hat{\mathbf{h}}_n^{stud}$, the distillation loss is

$$L_{distill} = \sum_{n} p_n \cdot (\log p_n - \log q_n) \tag{8}$$

$$\log p_n = \frac{\exp(((\hat{\mathbf{h}}_n^{tech})^T \cdot \hat{\mathbf{h}}_n^{tech})/\tau)}{\sum_n \exp(((\hat{\mathbf{h}}_n^{tech})^T \cdot \hat{\mathbf{h}}_n^{tech})/\tau)} \tag{9}$$

$$\log q_n = \frac{\exp(((\hat{\mathbf{h}}_n^{stud})^T \cdot \hat{\mathbf{h}}_n^{stud})/\tau)}{\sum_n \exp(((\hat{\mathbf{h}}_n^{stud})^T \cdot \hat{\mathbf{h}}_n^{stud})/\tau)} \tag{10}$$

where $p_n$ and $q_n$ denote the probability distributions of the corresponding sample $n$ in the teacher and student models, respectively, and $\tau$ is the temperature parameter in the knowledge distillation. After training with this loss, the student model will acquire a representation capability close to that of the teacher model, but with a more efficient inference speed.

## 4. Experiment and Analysis

### *4.1. Datasets and Metrics*

**MIR Flickr-25K** [40]: In the MIR Flickr dataset, there are 25,000 images and 38 concepts of ground truth labels. In our experiments, we selected images with at least 1 of the 38 labels. Thus, a total of 16,000 images were used for training and 2000 images for testing.

**NUS-WIDE** [41]: It is a large-scale dataset that can be used to evaluate multiple multimedia tasks. Much of their image data come from contributions from social media sites. The dataset contains 269,648 images and 81 manual labels as ground truth that can be used for performance evaluation. It also contains 5018 tags annotated by amateur users. In our experiments, we only used the 21 most common labels and all images associated with these labels. Thus, we form a training set of 100,000 images and a test set of 2000 images.

In our experiments, the Hamming distances of the query images and the images in the training set are used for ranking. We consider a correct retrieval result when there is an identification label in the query image and the returned image. Evaluation metrics include mean accuracy (MAP), precision, and recall.

### *4.2. Training*

Our backbone is pre-trained on ImageNet. We trained our model including a teacher module and a student module using mini-batch gradient descent with a learning rate of 0.001 and a learning rate of 0.0001 to fine-tune the backbone. We also used the momentum term with the rate of momentum equal to 0.9. The weighing factors for the losses $\lambda_1$ and $\lambda_2$ are all set to 1.0 for all the experiments, which were determined by cross-validation.

### *4.3. Analysis of the Lifting Effect of Transformer and Distillation*

In this section, we provide an experimental study on using the Transformer model and fast student model for retrieval. To analyze the effectiveness of the Transformer model and fast student model, we report the experimental results of our method with different modules on MIRFlickr-25K and NUS-WIDE datasets. Our main findings are summarized below.

**Transformer teacher model is better than the student model in retrieval accuracy.** Table 1 compares the various variants of our approach. We observe that the attention-based Transformer model outperforms the linear projection-based fast student model. This suggests that the self-attentive mechanism in Transformer has the potential to improve the ability to discriminate features and thus improve retrieval quality.

**The student model has significant advantages over the teacher model in terms of retrieval efficiency.** As we can see from Table 1, although the Transformer-based teacher model achieves high accuracy, it is too slow to generate hash codes. On the other hand, the fast student model achieves nearly 10 times the computational speed of the teacher model. This is because Transformer is computationally intensive. There exists a large amount of computation in the self-attention process.

**Table 1.** Transformer-based teacher model and fast student model comparison. Length is the size of hash codes. Time indicates the time for the model to generate the hash codes on the query image.

| Model | Train Data | Length | MAP | Time |
|---|---|---|---|---|
| Teacher | MIRFlickr-25K | 24 | 0.7582 | 2.11 s |
| | | 48 | 0.7435 | 2.18 s |
| Student | | 24 | 0.7112 | 0.19 s |
| | | 48 | 0.7088 | 0.21 s |
| Teacher | NUS-WIDE | 24 | 0.6932 | 2.11 s |
| | | 48 | 0.6882 | 2.18 s |
| Student | | 24 | 0.6473 | 0.19 s |
| | | 48 | 0.6482 | 0.21 s |

**Distillation improves the student model.** In Table 2, we compare two methods on NUS-WIDE and MIRFlickr-25K datasets, one with no teacher guidance and directly using the loss functions Equations (6) and (7) to train the student model, and the other with teacher guidance for distillation training denoted as Transformer. As seen in Table 2, the distillation method improves the performance of the student model by more than 10% on MAP when trained on NUS-WIDE, which significantly reduces the gap between the slow and fast models. On the other hand, the improvement when trained on MIRFlickr-25K is not as significant as on NUS-WIDE, probably due to the more minor training data of MIRFlickr-25K. In addition, the teacher model has limited ability to generalize the knowledge learned on the small training data, and thus has a lower upper limit when distillation learning.

**Table 2.** Distillation experiment with the Transformer-based model as the teacher and the linear projection-based fast model as the student.

| Model | Teacher | Train Data | Length | MAP |
|---|---|---|---|---|
| Student | None | MIRFlickr-25K | 24 | 0.6678 |
| | Transformer | | 24 | 0.7112 |
| | None | | 48 | 0.6561 |
| | Transformer | | 48 | 0.7088 |
| Student | None | NUS-WIDE | 24 | 0.5321 |
| | Transformer | | 24 | 0.6473 |
| | None | | 48 | 0.5396 |
| | Transformer | | 48 | 0.6482 |

### 4.4. Comparison to the State of the Art

In this session, we compared our method with the state-of-the-art methods performed using the same evaluation metrics. The contrasting methods include LSH [42], ITQ [16], SH [15], PCAH [43], SpH [44], DH [45], DeepBit [46], DSH [47], BDNN [48], DVB [49] and DOH [50]. The LSH, ITQ, SH, PCAH, SpH, and DSH are not deep learning based hashing methods. We extracted depth features from the pre-trained ResNet model [27] and used them as input for these methods in order to make a fair comparison. For the deep learning-based methods, i.e., DH, DeepBit, BDNN, and DOH, we evaluated them with the hyperparameter settings suggested in their papers and ran the source codes provided by the authors. For DVB [49], we directly refer to the results of the original paper.

The MAP results calculated using different lengths of hash codes on two datasets, MIRFlickr-25K and NUS-WIDE, are reported in Table 3. The reported results show the superiority of the proposed method and validate that the motivation of the proposed method is valid. Furthermore, by incorporating the self-attention mechanism of the Transformer model, image hash codes can be made global and unique. At the same time, this advantage can be instilled into the efficient lightweight model by knowledge distillation, which reduces the computational effort and has the feature advantage of the Transformer.

**Table 3.** MAP results for the MIR Flickr-25K and NUS-WIDE datasets, using hash codes of different lengths, were calculated using the first 5000 images retrieved.

| Methods | MIR Flickr-25K | | | | NUS-WIDE | | | |
|---|---|---|---|---|---|---|---|---|
| Length | 12 | 24 | 32 | 48 | 12 | 24 | 32 | 48 |
| LSH [42] | 0.5763 | 0.6065 | 0.5966 | 0.6263 | 0.3523 | 0.4096 | 0.4186 | 0.4555 |
| SH [15] | 0.6621 | 0.6433 | 0.6296 | 0.6225 | 0.5652 | 0.5061 | 0.4866 | 0.4546 |
| SpH [44] | 0.5982 | 0.5832 | 0.5831 | 0.582 | 0.4656 | 0.4662 | 0.4473 | 0.4481 |
| ITQ [16] | 0.6932 | 0.7082 | 0.6686 | 0.6991 | 0.6332 | 0.6255 | 0.5922 | 0.6481 |

**Table 3.** *Cont.*

| Methods | MIR Flickr-25K | | | | NUS-WIDE | | | |
|---|---|---|---|---|---|---|---|---|
| PCAH [43] | 0.6444 | 0.6321 | 0.6377 | 0.6534 | 0.5775 | 0.5052 | 0.4921 | 0.4924 |
| DSH [47] | 0.6962 | 0.7076 | 0.6851 | 0.6612 | 0.5944 | 0.5987 | 0.5725 | 0.5795 |
| DH [45] | 0.6021 | 0.6176 | 0.6144 | 0.6174 | 0.4745 | 0.4631 | 0.4625 | 0.4755 |
| DeepBit [46] | 0.5887 | 0.6033 | 0.6092 | 0.6091 | 0.5465 | 0.5551 | 0.5626 | 0.5612 |
| BDNN [48] | 0.6654 | 0.6692 | 0.6678 | 0.6695 | 0.5932 | 0.5922 | 0.5912 | 0.6098 |
| DVB [49] | - | - | - | - | - | - | 0.562 | - |
| DOH [50] | - | - | 0.6728 | 0.6712 | - | - | 0.6145 | 0.6251 |
| Ours—student | 0.7046 | 0.7112 | 0.7092 | 0.7088 | 0.6361 | 0.6473 | 0.6526 | 0.6482 |
| Ours—teacher | 0.7471 | 0.7582 | 0.7485 | 0.7435 | 0.6818 | 0.6932 | 0.6925 | 0.6882 |

Figures 3 and 4 show the performance curves of retrieval results on the MIRFlickr-25K dataset and the NUS-WIDE dataset. It can be seen from these two figures that the proposed method outperforms all the compared methods on both datasets. The results express the superiority of our method over the compared methods.

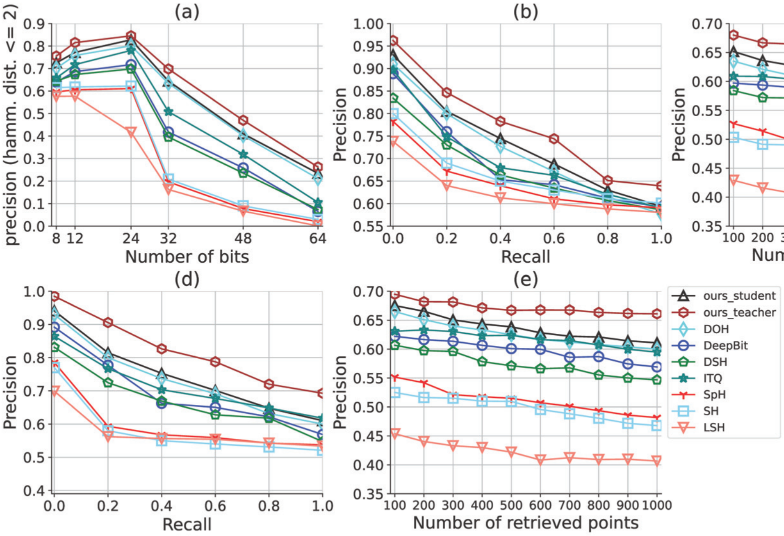


**Figure 3.** Performance curves of retrieval results on the MIRFlickr-25K dataset. (**a**) Precision using hash lookup within the Hamming radius 2; (**b**) Precision–recall curve for 48 bits; (**c**) Precision curve for 48 bits; (**d**) Precision–recall curve for 24 bits; and (**e**) Precision curve for 24 bits.

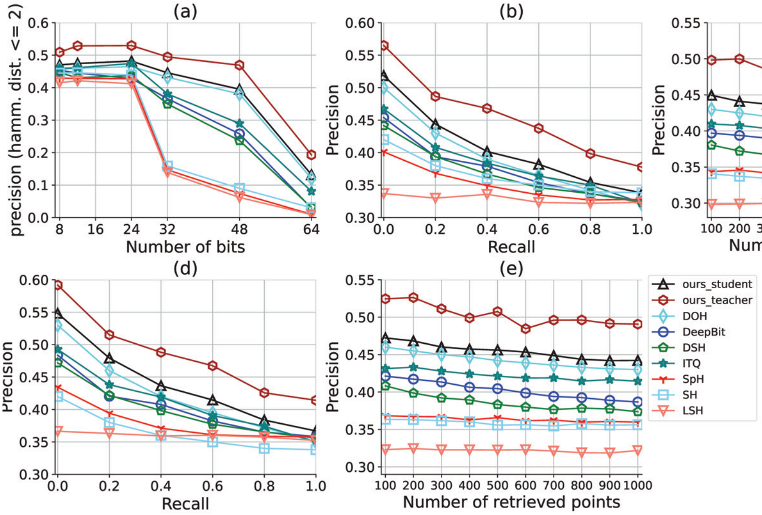


**Figure 4.** Performance curves of retrieval results on the NUS-WIDE dataset: (**a**) Precision using hash lookup within Hamming radius 2; (**b**) Precision–recall curve for 48 bits; (**c**) Precision curve for 48 bits; (**d**) Precision–recall curve for 24 bits; and (**e**) Precision curve for 24 bits.

## 5. Conclusions

This paper proposes a Transformer-based image hash learning with a knowledge distillation framework. By combining the self-attention mechanism of the Transformer model, the image hash code is enabled to be global and unique. At the same time, this advantage is instilled into the efficient lightweight model by knowledge distillation, thus reducing the computation and having the advantage of the Transformer's features. Experimental results on MIRFlickr-25K and NUS-WIDE datasets show that our approach can effectively improve the accuracy and efficiency of image retrieval.

**Author Contributions:** Funding acquisition, C.W. and W.Y. (Wujun Yang); Methodology, Y.L.; Resources, W.Y. (Wanteng Yuan); Writing—original draft, X.Q.; Writing—review & editing, W.Y. (Wujun Yang) and W.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the China University Industry-University-Research Innovation Fund (Grant No. 2021FNA03001), the Research Project on Postgraduate Education and Teaching Reform of Xi'an University of Posts and Telecommunications (Grant No. YJGJ202034), the fellowship of China Postdoctoral Science Foundation (Grant No. 2020M683695XB), Shaanxi Provincial Philosophy and Social Science Research Project in Major Theoretical and Practical Issues (Grant No. 2022ND0181), and Xi'an Social Science Planning Fund Project (Grant No. 22LW44).

**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. Birjandi, M.; Mohanna, F. 24 MODIFIED KEYWORD BASED RETRIEVAL ON FABRIC IMAGES. *Quantum J. Eng. Sci. Technol.* **2020**, *1*, 1–14.
2. Rout, N.K.; Atulkar, M.; Ahirwal, M.K. A review on content-based image retrieval system: Present trends and future challenges. *Int. J. Comput. Vis. Robot.* **2021**, *11*, 461–485. [CrossRef]
3. Peng, T.Q.; Li, F. Image retrieval based on deep convolutional neural networks and binary hashing learning. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 1742–1746.
4. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
5. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
6. Ribeiro, M.T.; Wu, T.; Guestrin, C.; Singh, S. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 4902–4912.
7. Bahdanau, D.; Cho, K.H.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
8. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
9. Hubel, D.H.; Wiesel, T.N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **1962**, *160*, 106. [CrossRef]
10. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in vision: A survey. In *ACM Computing Surveys (CSUR)*; Association for Computing Machinery (ACM): New York, NY, USA, 2021.
11. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
12. El-Nouby, A.; Neverova, N.; Laptev, I.; Jégou, H. Training vision transformers for image retrieval. *arXiv* **2021**, arXiv:2102.05644.
13. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
14. Gionis, A.; Indyk, P.; Motwani, R. Similarity search in high dimensions via hashing. In Proceedings of the International Conference on Very Large Data Bases, Scotland, UK, 7–10 September 1999; pp. 518–529.
15. Jain, P.; Kulis, B.; Grauman, K. Fast image search for learned metrics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
16. Liu, W.; Wang, J.; Ji, R.; Jiang, Y.G.; Chang, S.F. Supervised hashing with kernels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2074–2081.
17. Yang, H.F.; Lin, K.; Chen, C.S. Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 437–451. [CrossRef]
18. Zhao, W.; Luo, H.; Peng, J.; Fan, J. Spatial pyramid deep hashing for large-scale image retrieval. *Neurocomputing* **2017**, *243*, 166–173. [CrossRef]
19. Zhao, W.; Guan, Z.; Luo, H.; Peng, J.; Fan, J. Deep Multiple Instance Hashing for Fast Multi-Object Image Search. *IEEE Trans. Image Process.* **2021**, *30*, 7995–8007. [CrossRef]
20. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
21. 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, 20–25 June 2005; Volume 1, pp. 886–893.
22. Chugh, H.; Gupta, S.; Garg, M.; Gupta, D.; Mohamed, H.G.; Noya, I.D.; Singh, A.; Goyal, N. An Image Retrieval Framework Design Analysis Using Saliency Structure and Color Difference Histogram. *Sustainability* **2022**, *14*, 10357. [CrossRef]
23. Xia, R.; Pan, Y.; Lai, H.; Liu, C.; Yan, S. Supervised hashing for image retrieval via image representation learning. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014.
24. Zhang, R.; Lin, L.; Zhang, R.; Zuo, W.; Zhang, L. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE Trans. Image Process.* **2015**, *24*, 4766–4779. [CrossRef]
25. Kaur, P.; Harnal, S.; Tiwari, R.; Alharithi, F.S.; Almulihi, A.H.; Noya, I.D.; Goyal, N. A hybrid convolutional neural network model for diagnosis of COVID-19 using chest X-ray images. *Int. J. Environ. Res. Public Health* **2021**, *18*, 12191. [CrossRef]
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5999–6009.
27. 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.
28. Chen, M.; Radford, A.; Child, R.; Wu, J.; Jun, H.; Luan, D.; Sutskever, I. Generative pretraining from pixels. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 1691–1703.
29. Ullman, S. *High-Level Vision: Object Recognition and Visual Cognition*; MIT Press: Cambridge, MA, USA, 2000.

30. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
31. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai J F, D.D. Deformable transformers for end-to-end object detection. In Proceedings of the 9th International Conference on Learning Representations, Virtual Event, Austria, 3–7 May 2021.
32. Wang, H.; Zhu, Y.; Adam, H.; Yuille, A.; Chen, L.C. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 5463–5474.
33. Lu, X.; Zhao, T.; Lee, K. VisualSparta: An Embarrassingly Simple Approach to Large-scale Text-to-Image Search with Weighted Bag-of-words. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Virtual Event, 1–6 August 2021; pp. 5020–5029.
34. Henkel, C. Efficient large-scale image retrieval with deep feature orthogonality and Hybrid-Swin-Transformers. *arXiv* **2021**, arXiv:2110.03786.
35. Yang, M.; He, D.; Fan, M.; Shi, B.; Xue, X.; Li, F.; Ding, E.; Huang, J. Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 11772–11781.
36. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin transformer v2: Scaling up capacity and resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022; pp. 12009–12019.
37. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv* **2014**, arXiv:1412.6550.
38. Mirzadeh, S.I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 5191–5198.
39. Tian, J.; Xu, X.; Shen, F.; Yang, Y.; Shen, H.T. TVT: Three-Way Vision Transformer through Multi-Modal Hypersphere Learning for Zero-Shot Sketch-Based Image Retrieval. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 22 February–1 March 2022; Volume 36, pp. 2370–2378. [CrossRef]
40. Huiskes, M.J.; Thomee, B.; Lew, M.S. New Trends and Ideas in Visual Concept Detection: The MIR Flickr Retrieval Evaluation Initiative. In Proceedings of the ACM International Conference on Multimedia Information Retrieval, Philadelphia, PA, USA, 29–31 March 2010; pp. 527–536.
41. Chua, T.S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; Zheng, Y. NUS-WIDE: A real-world web image database from National University of Singapore. In Proceedings of the ACM International Conference on Image and Video Retrieval, Santorini Island, Greece, 8–10 July 2009; pp. 1–9.
42. Charikar, M.S. Similarity estimation techniques from rounding algorithms. In Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing, Montreal, QC, Canada, 19–21 May 2002; pp. 380–388.
43. Wang, J.; Kumar, S.; Chang, S.F. Semi-Supervised Hashing for Large-Scale Search. *IEEE Trans. Pattern. Anal. Mach. Intell.* **2012**, *34*, 2393–2406. [CrossRef] [PubMed]
44. Sun, A.; Bhowmick, S.S. Quantifying tag representativeness of visual content of social images. In Proceedings of the ACM International Conference on Multimedia, Firenze, Italy, 25 October 2010; pp. 471–480.
45. Liong, V.E.; Lu, J.; Wang, G.; Moulin, P.; Zhou, J. Deep hashing for compact binary codes learning. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
46. Lin, K.; Lu, J.; Chen, C.S.; Zhou, J. Learning Compact Binary Descriptors with Unsupervised Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
47. Jin, Z.; Li, C.; Lin, Y.; Cai, D. Density Sensitive Hashing. *IEEE Trans. Cybern.* **2017**, *44*, 1362–1371. [CrossRef]
48. Do, T.T.; Doan, A.D.; Cheung, N.M. Learning to Hash with Binary Deep Neural Network. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016.
49. Shen, Y.; Liu, L.; Shao, L. Unsupervised binary representation learning with deep variational networks. *Int. J. Comput. Vis.* **2019**, *127*, 1614–1628. [CrossRef]
50. Xie, F.; Zhao, W.; Guan, Z.; Wang, H.; Duan, Q. Deep objectness hashing using large weakly tagged photos. *Neurocomputing* **2022**, *502*, 186–195. [CrossRef]

*electronics*

MDPI

*Article*

# Multi-Scale Convolution-Capsule Network for Crop Insect Pest Recognition

**Cong Xu *, Changqing Yu, Shanwen Zhang and Xuqi Wang**

College of Information Engineering, Xijing University, Xi'an 710123, China; xaycq@163.com (C.Y.); wjdw716@163.com (S.Z.); 18811166421@163.com (X.W.)

* Correspondence: xucong0623@126.com

**Abstract:** Accurate crop insect pest identification in fields is useful to control pests and beneficial to agricultural yield and quality. However, it is a difficult and challenging problem due to the crop insect pests being small with various sizes, postures, shapes, and disorganized backgrounds. Multi-scale convolution-capsule network (MSCCN) is constructed for crop insect pest identification. It consists of a multi-scale convolution module, capsule network (CapsNet) module, and SoftMax classification module. Multi-scale convolution is used to extract the multi-scale discriminative features, CapsNet is employed to encode the hierarchical structure of the size-variant insect pests in the crop images, and Softmax is adopted for insect pest identification. MSCCN combines the advantages of convolutional neural network (CNN), CapsNet, and multi-scale CNN, and can learn multi-scale robust features from pest images of different shapes and sizes for pest recognition and identify various morphed pests. Experimental results on the crop pest image dataset show that this method has a good recognition rate of 91.4%.

**Keywords:** crop insect pest identification; convolutional neural network (CNN); capsule network (CapsNet); multi-scale convolution-capsule network (MSCCN)

**Citation:** Xu, C.; Yu, C.; Zhang, S.; Wang, X. Multi-Scale Convolution-Capsule Network for Crop Insect Pest Recognition. *Electronics* **2022**, *11*, 1630. https://doi.org/10.3390/electronics11101630

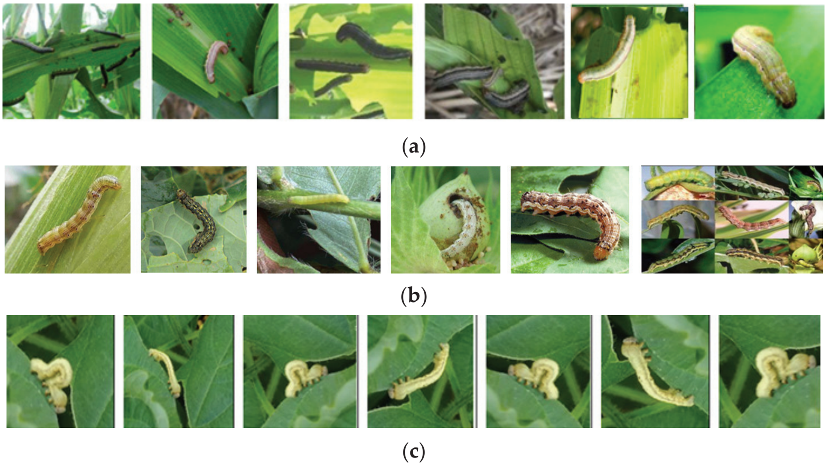
Academic Editor: Javid Taheri

Received: 9 April 2022
Accepted: 12 May 2022
Published: 20 May 2022

## 1. Introduction

To control pests, avoid economic losses, and reduce pesticide costs, early detection and identification of crop pests is an important task. However, it is difficult and challenging to detect and recognize crop pests in fields, because the insect pest images are photographed in complex crop environments. These include not only various types, sizes, postures, and shapes of insect pests, but changeable light, viewpoint, and irregular backgrounds, and it is obvious that the insect pest size is small in proportion to the whole image and its color and texture characteristics are similar to those of the background in the cropped image, as shown in Figure 1. Therefore, it usually leads to low identification accuracy using the traditional pattern recognition and image processing algorithms [1].

With the development of computer vision technology, computer computing power, and various algorithms of artificial intelligence (AI) [1,2], machine learning [3,4], and modern digital and deep learning [5], many crop pest detection and recognition methods have been presented [6]. Martineau et al. [7] investigated forty-four studies on this topic, including a lot of methods of image capture, feature extraction, and classification and tested datasets, and generally discussed the questions that might still remain unsolved. Costa et al. [8] constructed a knowledge-based crop pest identification system. This system can provide a convenient way for farmers to manage crop pests and diseases. Liu et al. [9] introduced the definition and connotation of the crop disease-pest knowledge and analyzed and classified the key techniques and methods of crop disease-pest detection and recognition in recent years, including knowledge representation, feature extraction and fusion, reasoning, and classifier. Huo et al. [10] introduced the research progress of disease-pest identification, pest number, and position detection, of an existing dataset and some methods used in previous articles. Li et al. [11] proposed a few-shot cotton pest recognition method and verified its effectiveness and feasibility on two datasets, namely the national bureau of agricultural insect resources and a dataset with the natural scenes. The results of the above methods show that the performance of the traditional pest identification methods relies on hand-crafted features and matching templates, shallow learning-based features with limited representation power, and only low-level features, but ignores the hierarchical features of pest images, so their recognition rate and generalization ability are limited.


(**a**)

(**b**)

(**c**)

**Figure 1.** Crop insect pest image examples. (**a**) Maize army worm. (**b**) Cotton bollworm. (**c**) Bean larvae.

Convolutional neural network (CNN) has made remarkable achievements in various target detection and recognition tasks. It has been widely used in pest detection and recognition as it can automatically learn the essential features of the pest images from a large amount of data and produce fewer high-quality candidate features for pest recognition [12]. Ai et al. [13] used CNN to automatically identify crop disease pests as they trained the Inception-ResNet-v2 model, utilizing the public dataset of the AI Challenger Competition in 2018, with 27 disease images of 10 crops, and designed and implemented the Wechat applet of crop disease-insect pest recognition. Xie et al. [14] proposed an automatic crop pest classification method by learning multi-level features from a large number of unlabeled image patches using unsupervised feature learning methods and utilized the filters in multiple scales coupling them with several pooling granularities. Labaa et al. [15] proposed a crop pest recognition method based on CNN, improved by combining different technologies such as CNNs and REST services. Li et al. [16] proposed a fine-tuned GoogLeNet model to deal with the complicated backgrounds presented by farmland scenes and achieved better pest classification results than the original model.

Compared to traditional handcraft-feature extraction algorithms, CNN is effective in image classification tasks. It can automatically learn features during the training process, avoiding the error generated by manual selection, but its pooling operation (down-sampling) can only give rough location information, allowing the model to ignore some small spatial changes and failing to accurately learn the location association of different objects, such as the location, size, direction, and even deformation degree and texture of entities in a region. Although the pooling operation of CNN can maintain the invariability of the location and direction of the entity, it will lose the characteristics of small pests, so the recognition rate of crop pests may not be high. Therefore, pooling operations may cause some problems: they may lose the low-level features and spatial hierarchical features, and the data of small pests (under certain conditions) may be lost after down-sampling.

The capsule network (CapsNet) is a new kind of deep learning architecture aiming to encode the features of the images and their spatial relationships [17]. It can overcome the shortcomings of CNN. It only uses the shallow CNN to preserve the spatial information, and can capture not only the discriminant features, but also the underlying relationships between these features. A capsule is a group of neurons whose output represents the various perspectives of an entity, such as pose, texture, scale, or the relative relationship between the entity and its parts. In this case, CapsNet is more robust to affine transformations and achieves good results with fewer training samples. Paoletti et al. [18] constructed a CapsNet for hyperspectral image classification, where several spectral-spatial capsules are used to learn HSI spectral-spatial features while significantly reducing the network complexity. Mensah et al. [19] proposed Gabor CapsNet for plant disease detection and evaluated its performance on three publicly available plant disease datasets containing disease leaf images with high similarity and background objects. Wang et al. [20] proposed a multi-scale convolutional CapsNet for hyperspectral image classification, which is composed of a multi-scale convolutional layer, a single-scale convolutional layer, a PrimaryCaps layer, a DigitCaps layer, and a fully connected layer. Peker [21] proposed a multi-channel CapsNet ensemble for plant disease detection and individually trained the network on the image set. Thenmozhi et al. [22] proposed a deep CNN model to classify insects, where transfer learning was applied to fine-tune the pre-trained models.

From the above analysis, it is known that the conventional CNN-based crop pest image classification faces a problem of quite limited training samples, which leads to overfitting and dissatisfied performance to describe the correlation between features. CapsNet can deal with the disadvantages of CNN, but the feature representation capability of the low-level features extracted by the shallow-layer CNN is limited. Therefore, the original CNN or CapsNet is not suitable for crop pest recognition tasks. Inspired by multi-scale convolutional CapsNet and multi-channel CapsNet, a multi-scale convolution-capsule network (MSCCN) is constructed for crop insect pest recognition combining the advantages of traditional CNN and CapsNet. It consists of a multi-scale convolutional module, CapsNet module, and a Softmax classification module. The main contributions of this work are as follows:

1. Inception is introduced into the convolutional module with different-scale convolutional kernels in different branches of the Inception structure, and multi-scale image features are extracted by different receptive fields in each branch, which increases the width of the network and the adaptability of the network to pest scale;
2. An improved dropout is proposed on the encoded capsules to enhance the robustness of the model for the capsule layer.

The remainder of the paper is organized as follows. Section 2 reviews the related works including Inception and CapsNet. MSCCN is introduced in detail in Section 3. Experiments are presented in Section 4. Section 5 concludes the paper and puts forward some opinions and suggestions for the future research direction.

## 2. Related Methods

### *2.1. Inception*

Inception is a module in GoogleNet and has been validated to be better in complex images classification tasks. It has multi-scale convolution kernels to extract the features of different scales from the input images by increasing the number of convolutional kernels and introducing multi-scale convolutional kernels. The inception structure has been improved in terms of speed and accuracy. There are multiple versions of Inception: Inception V1, Inception V2, Inception V3, Inception V4, and Inception ResNet, each of which is an iterative evolution of the previous version. In general, a lower version of the Inception module may work better in classification tasks. Figure 2 shows Inception V1. As shown in Figure 2, 1 × 1, 3 × 3, and 5 × 5 convolutional kernels are used to convolve the outputs of the upper layer at the same time to form a multi-branch structure. Feature maps obtained from the different branches are then concatenated to obtain different classification features of the input images. Processing these operations in parallel and combining all the results will result in better image representation. To make the feature map have the same size, each branch adopts the same padding mode with the stride of 1. The 1 × 1 convolution operation is used before 3 × 3 and 5 × 5 and after Max-pooling to reduce the amount of calculation.

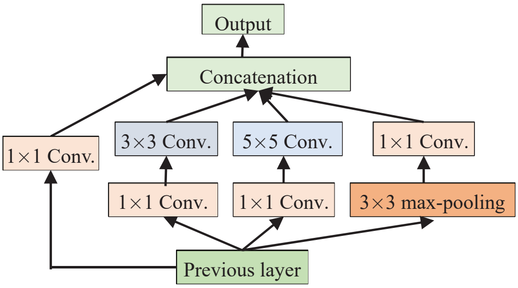


**Figure 2.** The structure of Inception V1.

*2.2. Capsule Network (CapsNet)*

CNN is composed of multiple neurons stacked together, and it takes a lot of computation to compute convolution between neurons, so the pooling operation is used to reduce the size of the network layer. However, classification information may be lost by pooling. CapsNet is constructed to overcome the limitations and shortcomings of CNN. It can encode spatial information and calculate the existence probability of objects, and is good at dealing with changeable object recognition with different positions, sizes, directions, deformations, speeds, textures, and other features. Its architecture is shown in Figure 3, consisting of a traditional convolution layer, a primary capsule layer, and a digital capsule layer.

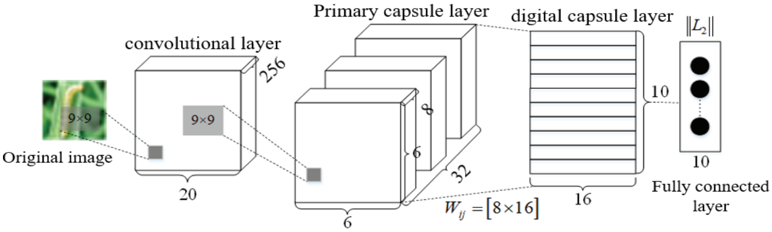


**Figure 3.** The architecture of CapsNet.

In CapsNet, the primary capsule layer mainly transforms the upper scalar representation into a vector representation, so its output is as a vector. The digital capsule uses dynamic routing algorithms to update the network. The final output is vectors. The length of each vector is the probability value of belonging to a class.

## 3. Multi-Scale Convolution-Capsule Network (MSCCN)

Motivated by the fact that the crop insect pests are changeable with various postures, and their sizes range from less than 1 mm to more than 100 mm, a multi-scale convolution-capsule network (MSCCN) is proposed for crop insect pest recognition. Its architecture is shown in Figure 4.

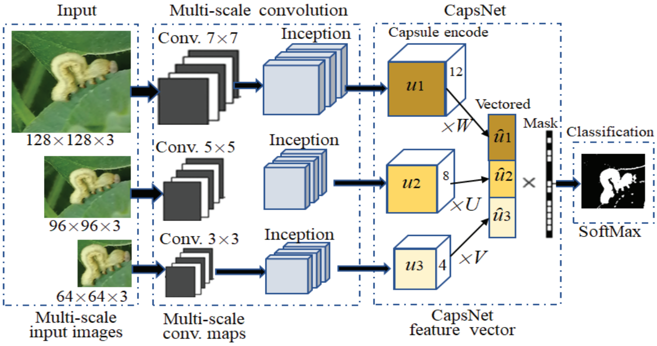


**Figure 4.** Architecture of MSCCN.

The input image is reshaped to 128 × 128, 96 × 96, and 64 × 64 assembled in parallel. MCNN firstly extracts the high-level features of describing pest images through three multi-scale convolutions, three Inceptions, and three CapsNet using these features to further construct the vector-based capsule structure to form the final discriminative feature vector of pests in the image, which will be directly fed to the final SoftMax classifier without any feature reduction. Finally, pest recognition is implemented by the Softmax classifier. MCNN is designed as an end-to-end structure for easy convolution-CapsNet training and deployment.

In CapsNet, three multi-dimensional primary capsules are employed to encode the hierarchical multi-scale features extracted by three multi-scale convolutions, and obtain 12D, 8D, and 4D capsules, respectively. Then, the predicted vectors are computed through different weight matrixes $W$, $V$, and $U$ as follows:

$$\begin{aligned} \overset{\frown}{u}{}^{1}_{j|i} &= W{\cdot}u^1_i \\ \overset{\frown}{u}{}^{2}_{j|i} &= V{\cdot}u^2_i \\ \overset{\frown}{u}{}^{3}_{j|i} &= U{\cdot}u^2_i \end{aligned} \tag{1}$$

where $u^1, u^2, u^3$ are the feature maps of three multi-scale convolutions, $W, V$, and $U$ are three weight matrixes of $u^1, u^2, u^3$ and $\overset{\frown}{u}{}^1, \overset{\frown}{u}{}^2, \overset{\frown}{u}{}^3$ respectively, $u^k_i$ is $i$-th primary-capsule from $k$-th branch, $\hat{u}^k_{j|i}$ is predict vector between the $j$-th parent capsule and the $i$-th child capsule of $k$-th branch, and $\hat{u}$ is the output of this multi-scale capsule encoding structure, which concatenates the results of three branches by function *concat*().

The classification features are encoded using a weight matrix between $i$-th child capsule and $j$-th parent capsule. During the training, the part–whole relationship for each capsule pair is learned by adjusting the transformation matrixes $W, V$, and $U$.

There is a dynamic routing between the multi-scale capsule encoding unit and digit capsule layer. It is used to ensure that the outputs of child capsules are sent to the proper parent capsules. The prediction vectors $\hat{u}$ in the previous section are computed through a weight matrix. The relationship is determined between each parent capsule $s_j$ and prediction vector $\hat{u}$ by dynamic routing. All the prediction vectors are denoted as $\hat{u}_{j|i}(i = 1, \cdots, n)$. In the first iteration, $c^1_i = \frac{1}{n}$ and $s^1_j = \sum_{i=1}^{n} c^1_i \hat{u}_{j|i}$, where $\sum_j c_j = 1$ and $c_j \geq 0$. Then, adjust the routing coefficients $c^1$ to $c^2$ by the function *update*() as follows:

$$\begin{aligned} b^{i+1} &= b^i + \hat{u}_j{\cdot}v_j \\ c^{i+1} &= soft\max(b^{i+1}) \end{aligned} \tag{2}$$

where $b$ is the coupling coefficient before normalization and $b^1 = 0$, $v_j$ is the $j$th output capsule of the parent capsule layer calculated by

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \cdot \frac{s_j}{\|s_j\|} \tag{3}$$

where $s_j$ is the total input vector of the $j$th capsule obtained by the weighted sum of the $j$th parent capsule layer connecting with the $i$th child capsule layer, $\frac{\|s_j\|^2}{1+\|s_j\|^2}$ is the reduction coefficient of $s_j$, $\frac{s_j}{\|s_j\|}$ is the normalized unit vector of $s_j$, $s_j = \sum\limits_i c_{ij}\hat{u}_{j|i}$, and the prediction vector $\hat{u}_{j|i}$ is obtained by multiplying the output features of the BN layer with the weight matrix of the primary capsule layer.

The objective function of MCNN is expressed as follows:

$$L_c = \sum_{k \in CNum} T_k \max(0, m^+ - \|V_k\|^2) + \lambda(1 - T_k)\max(0, \|V_k\| - m^-)^2 \tag{4}$$

where the former part is used to calculate the settings of the correctly classified digital capsule, the latter part is used to calculate the losses of wrongly classified digital capsules, $m^+$ = 0.9 and $m^-$ = 0.1 are the default category prediction values, $\lambda$ = 0.5 is the default balance coefficient, $T_k$ is the label of data category, $T_k = 1$ is the correct label, $T_k = 0$ is the incorrect label, CNum is the number of categories, $\|V_k\|$ is the length of the vector representing the probability of discriminating as the $k$th class pest, and the total loss is the sum of all digital capsule loss functions.

The main processes of MCNN-based crop pest recognition are shown in Figure 5.

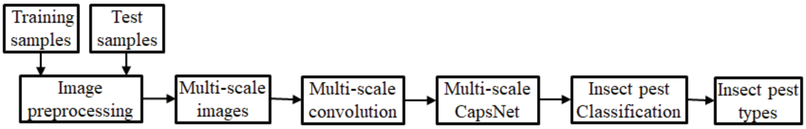


**Figure 5.** Crop pest recognition process based MSCCN.

First, all kinds of pest images are divided into the training set and test set. Both sets need to be preprocessed to facilitate MSCCN to extract the multi-scale features. Then, the results of image preprocessing are used as the input of the multi-scale convolution, the network will automatically extract the multi-scale features of color, texture, and shape from training samples. Multi-scale CapsNet is used to encode the multi-scale convolution features. Each layer of CapsNet is composed of neurons. The neuron input of CapsNet is vectors. The vector length represents the approximate probability of the pest. The vector direction represents the instantiation parameters of the pest. The output of a capsule is only routed to the next layer of the corresponding capsule, which will return a clearer input signal, it can accurately determine the posture of the pest. The feature combination method is adopted for different feature vectors. MSCCN structure and network parameters are set up. After training, the classification model is implemented to classify and recognize pest images by Softmax. The $k$-dimension feature vector $Y_i$ extracted by CapsNet is input into the trained Softmax classifier, as follows:

$$P(Y = i|x) = Soft\max(Y_i) = \frac{exp(\varpi_i Y_i)}{\sum\limits_{i=1}^{K} exp(\varpi_k Y_k)} \tag{5}$$

where $P$ is the probability that the feature vector $x$ belongs to the $i$th category, $K$ is the total number of categories, and $\varpi$ is the weight items.

From the above analysis, the pseudocode MSCCN is introduced in Algorithm 1:

**Algorithm 1:** Multi-scale CapsNet

```
Input training pest images, parameters: $\eta = 0.001$, $\alpha = 0.9$, $\beta = 0.99$, and $\varepsilon = 0.00001$,
batch-size = 128, the number of iterations and dynamic routing are 3000 and 3, respectively,
threshold $\rho$;
1: Image processing;
2: Reshape each image into three images with different sizes;
3: For iteration;
4: for $k$ = 1 to 3
    Carry on the $k$th convolution with different kernels with sizes of $7 \times 7$, $5 \times 5$ and $3 \times 3$;
    Carry on the $k$th Inception convolution;
    Carry on procedure routing ($k$, $\widehat{u}_{j|i}$, $r$, $l$) of the $k$th CapsNet;
    (1) for all capsule $i$ in layer $l$ and capsule $j$ in layer(+1): $b_{ij} \leftarrow 0$;
    (2) for $r$ iterations do
    for all capsule $i$ in layer $l$: $c_i \leftarrow$ softmax($b_i$)
    for all capsule $j$ in layer ($l$ + 1): $s_j \leftarrow \sum_j c_{ij} \widehat{u}_{j|i}$ by Equation (5)
    for all capsule $j$ in layer ($l$ + 1): $v_j \leftarrow squash(s_j)$
    (3) for all capsule $i$ in layer $l$ and capsule $j$ in layer ($l$ + 1): $b_{ij} \leftarrow b_{ij} + \widehat{u}_{j|i} v_j$
    (4) return $v_j$
    return $v_k$;
5: integrate $v_k$;
6: $v_k \times$ Mask;
7: input $v_k \times$ Mask into Softmax classifier;
8: calculate loss Lc by Equation (4);
9: if Lc more than $\rho$, return step 3;
10: Stop iterations.
```

## 4. Experiments

To evaluate the performance of the proposed method based on MSCCN, a lot of experiments were conducted on the rice pest image set of IP102 dataset and compared with four existing mainstream deep learning methods, AlexNet [12,23,24], CapsNet [19], MS-CapsNet [20], DCNN + transfer learning (DCNNTL) [22], and ResNet50 [25]. AlexNet consists of five convolutional layers, three max pooling layers, and three fully connected layers. Resnet50 is composed of 49 convolutional layers and a fully connected layer, where the residual network unit contains cross-layer connections. MS-CapsNet consists of a multi-scale convolutional layer, a single-scale convolutional layer, a primaryCaps layer, a digitCaps layer, and a fully connected layer. DCNNTL consists of six convolutional layers, five max pooling layers, and a fully connected layer. It uses VGG16 as transfer learning to pre-train the deep CNN model on the constructed dataset. In all models, categorical cross entropy is used as a loss function, Stochastic gradient descent (SGD) is used as the optimizer, and Softmax classifier is used in their output layers to classify pest categories.

The hardware and software conditions of the experiments are as follows: the operating system is 64-bit Microsoft Windows 10, the CPU is I5-6200U, GeForce RTX2080 SUPER 8GB, 64-bit Operating System and x64-based processor NVIDIA Ge Force RTx 2080Ti 11GB GDDR6 Mother Board Intel i7/i8/i9, the programming language is Python 3.7 Jupyter notebook software, and the deep learning framework is Keras 2.3.0.

### 4.1. IP102 Dataset

IP102 (https://github.com/xpwu95/IP102 (accessed on 7 April 2019)) is often used to test insect pest detection and recognition methods based on deep learning [26]. It contains 75,222 images belonging to 102 common crop insect pest categories with an average of 737 images per class. Most images were collected by common image search engines at different growth stages, and about 19,000 images were annotated with bounding boxes for pest detection. Some images are shown in Figure 6. From Figure 6, it is found that the insect pest images were collected in the fields with various sizes, shapes, and complex backgrounds, and the pest has different sizes, postures, and shapes at the different stages of the life cycle [23].

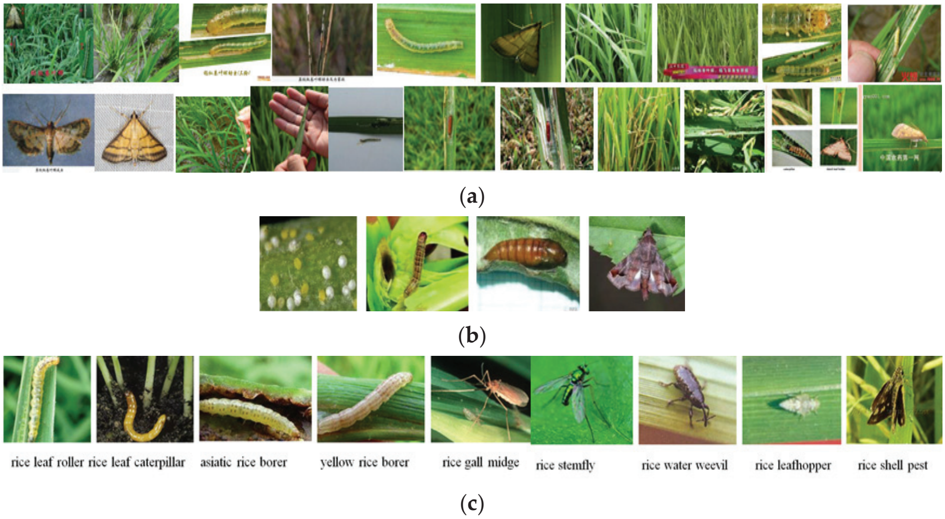


**Figure 6.** Insect pest image examples in IP102. (**a**) The first 20 original images. (**b**) Different forms of a kind of insect pest at different stages of the life cycle. (**c**) Nine kinds of rice annotated images.

IP102 contains 5701 original images of nine rice pests. Their pest image names and corresponding numbers and serial numbers are shown in Table 1, where the maximum number is 1115, and the minimum number is 369. The number of each kind of rice pest image is increased to more than 500 by the augmentation algorithm. All images are converted to JPEG format. Repeated or damaged images are deleted. In this study, all rice insect pest images are used to conduct rice insect pest recognition experiments, where each original image is firstly reshaped to 128 × 128, closer to the actual application, because the sizes of the collected images are not uniform. Finally, 5000 preprocessed images are used for experiments, except for 1034 poor quality or negative images.

**Table 1.** The number of rice pest images in IP102 dataset.

| No. | Pest Name | Number | Label |
|---|---|---|---|
| 1 | Rice leaf roller | 1115 | 0–1115 |
| 2 | Rice leaf caterpillar | 485 | 1116–1601 |
| 3 | Asiatic rice borer | 1053 | 1863–2915 |
| 4 | Yellow rice borer | 504 | 2916–3419 |
| 5 | Rice gall midge | 506 | 3420–3925 |
| 6 | Rice Stemfly | 369 | 3926–4294 |
| 7 | Rice water weevil | 856 | 6575–7430 |
| 8 | Rice leafhopper | 404 | 7431–7834 |
| 9 | Rice shell pest | 409 | 8008–8416 |
| | | Total | 5701 |

Due to the different crop pest conditions of data collection, illumination, and parameter settings of a digital scanner, color differences of digital pest images are often caused. Size and color normalization can not only ensure the color consistency of the original image, but also preserve the biological information in the pest image, so as to improve the recognition performance of the model. As the pest image sizes of the dataset are different, ranging from 220 × 220 to 512 × 512, the size of the input original image and ROI label will be uniformly adjusted to 128 × 128. At the same time, pixel values of all images will be regularized to between 0 and 255 when entering the channel expansion module. After channel expansion, minimax normalization is applied to the pest images of each channel to normalize the range of pixel values to between 0 and 1, so as to complete channel expansion of original pest images and better meet the input of the deep learning network. Minimax normalization is defined as follows,

$$x_{nor} = \frac{x - X_{\min}}{X_{\max} - X_{\min}} \tag{6}$$

where $x$ is the pixel value of the original image, $X_{\min}$ is the minimum value of the pixel value set, and $X_{\max}$ represents the maximum value of the pixel value set.

The 5-fold-cross-validation (5FCV) strategy is used to evaluate the performance of the proposed model. The 5FCV experiment is conducted 50 times, and the results are the average of 50 5FCV experiments.

### *4.2. Experiment Results*

Rice pest recognition mainly relies on MSCCN to extract the pest image features and complete the pest identification via Softmax classifier. MSCCN is a deep learning algorithm. The CapsNet module in MSCCN uses activity vectors to represent instantiation parameters of specific pest types. The length of the output vectors is used to characterize the probability of pests having the current input. After the pest images are preprocessed, the images are output into MSCCN, where CapsNet uses multi-scale convolution to extract the pest image features, trains the image classification, and predicts the output vector based on routing-by-agreement protocol. In this paper, the characteristics of MSCCN are used to solve the problem of crop pest identification difficulties caused by multiple pests overlapping during the pest recognition process. The most vital step, according to routing-by-agreement, is to analyze the pest images with overlapped objects. The parameters of MSCCN are originally set as a batch size 128, weight decay factor 0.00001, and number of training epochs 100, and the initial weights are set randomly from a Gaussian distribution with a mean of 0 and a variance of 1. In the multi-scale convolution module, the dropout rate is set to 0.4, the learning rate is initialized to $1 \times 10^{-3}$, decreasing 0.05 times as the number of iterations increases. In CapsNet module, the number of iterations of dynamic routing is 3000, and the dropout ratio is 0.9. Adam is employed as the gradient descent algorithm to perform the training. In Adam, the original parameters are set as $\eta = 0.001$, $\alpha = 0.9$, $\beta = 0.99$, and $\varepsilon = 0.00001$.

Three parameters in MSCCN are not trainable but can be fine-tuned: dropout rate, learning rate, and mini-batch size. They are fixed at the start of training. Considering validation accuracy while tuning hyperparameters, we fine-tune them. The dropout is used to reduce overfitting, and a dropout layer is often added after each dense layer except the last. MSCCN is trained with three dropout rates of 0.3, 0.4, and 0.5. The results are 0.891, 0.575, and 0.843, respectively. The dropout rate of 0.3 has the best result in general. In the experiments, the dropout rate is set to 0.3, which means that MSCCN model will randomly ignore 30% of the neurons of the previous layer. Learning rate determines how fast the weights of MSCCN are adjusted to find the local or global minima of the loss function. MSCCN is tested with a learning rate of 0.01, 0.001, 0.0001, and 0.00001. In terms of convergence speed and accuracy, learning rate of 0.0001 has the best accuracy of 0.906. MSCCN is evaluated with a mini-batch size of 10, 16, 32, 64, and 128. The accuracy of MSCCN is improved with the increase in mini-batch size from 10 to 64, and then it decreased for 128, as shown in Figure 7. Then, a mini-batch size of 64 is selected to train the model that increases the convergence precision.

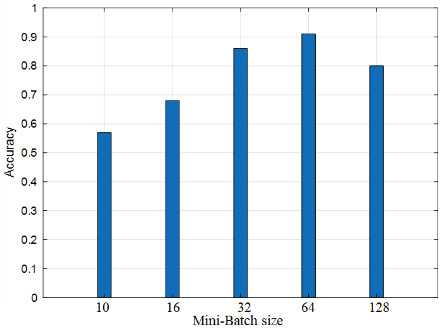


**Figure 7.** Accuracy versus Mini-Batch sizes.

Gradient descent and backpropagation algorithms are used to update the weight parameters of the model. As the gradient descent algorithm of the driving quantity is used, the momentum factor is set as 0.9 to prevent the overfitting problem. In order to show the performance of MSCCN, MSCCN is compared with classical CNN and CapsNet, and three modified models: MS-CapsNet, DCNNTL, and ResNet50.

Figure 8 shows the loss values of six models in the training set. From Figure 8, it is found that MSCCN converges fastest, and the curve is relatively stable after 2000 iterations. All models converge basically when the number of iterations is more than 2000. For fair comparison, in the following experiments, all trained models are selected after 3000 iterations to recognize pest categories. Table 2 shows the average recognition rates of 50 5FCV experiments by six models based on pest recognition methods.

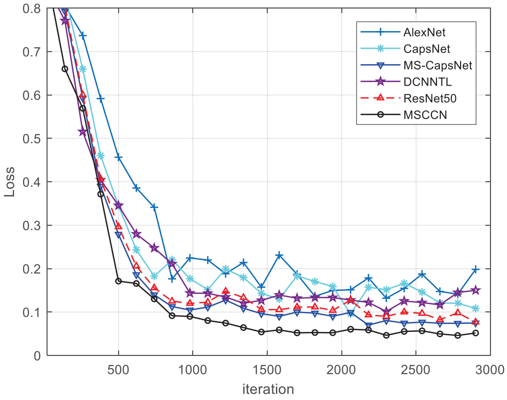


**Figure 8.** The loss versus iterations of three models.

**Table 2.** Average recognition rates of five models.

| Method | AlexNet | CapsNet | MS-CapsNet | DCNNTL | ResNet50 | MSCCN |
|---|---|---|---|---|---|---|
| Recognition rates | 0.803 | 0.824 | 0.896 | 0.847 | 0.855 | 0.924 |
| Training time (h) | 22.37 | 19.10 | 21.25 | 18.26 | 25.08 | 20.17 |
| Test time (s) | 0.35 | 0.30 | 0.37 | 0.28 | 0.29 | 0.26 |

### 4.3. Discussion

From Figure 8 and Table 2, it is found that MSCCN outperforms the other four models. Its generalization is enhanced because the multi-scale input, multi-scale convolution, and time-spatial characteristics can be extracted from various pest images, ultimately allowing the pest images to be characterized at a higher level of abstraction. The main reason is that MSCCN makes use of the advantages of multi-scale input, Inception, multi-scale CNN, and multi-scale CapsNet, so it can quickly extract the features from the pests with various sizes and shapes. MS-CapsNet is the second best because, similar to MSCCN, it employed multi-scale convolution to extract the image features of pests with scale changes and uses capsule network to extract the image features of pests with shape, position and angle changes. DCNNTL is better than AlexNet and CapsNet, because it employed transfer learning to speed up training. Though ResNet50 has the deepest layers, it does not work very well. The reason for this is that it needs a large number of training samples, but there are not enough samples. AlexNet has the worst performance, because it is difficult to optimize pest distortions at the same time by simply changing the size of the projection model, while AlexNet requires a large training database as a comparison library to improve its classification performance and overcome the overfitting problem. CapsNet is not very good because it has a shallow convolution layer, which cannot extract deep classification features.

The results in the references [12–16] verify that CNN and its variants are suitable for classifying images that are very close to the training data set, but they perform poorly in various pest images because pest images vary greatly. Pooling in CNN can establish invariance of location and size, but this invariance can also lead to objects with harmful colors and shapes being mistaken for pests. Humans can identify various pests through a few training images of pests, while CNN needs a large number of training samples, even tens of thousands, to train a good model, which is obviously too substantial. Unlike CNN, CapsNet extracts feature vectors, not feature maps. The vector modulus represents the probability of the feature existence, and the vector direction represents the attitude feature information. The moving features will change the CapsNet vector without affecting the probability of the features' existence. Therefore, CapsNet is more suitable to describe the characteristics of various pests. As CapsNet collects the pose information of pests, a good representation effect can be learned from a small number of samples, so the identification performance of pests is improved.

Unlike other deep models such as CNN and CapsNet, the components of MSCCN are intended to reveal typical time–spatial features and their corresponding instantiation parameters. These features allow the various pest images to be described at a higher level of abstraction while reducing the overfitting inherent in complex and deep networks.

## 5. Conclusions

Traditional pest identification methods cannot effectively extract robust classification features from the changeable images of pests. Many methods based on deep learning have great advantages in image recognition, but they require a large number of training samples and time for training parameters. To improve the recognition performance, a multi-scale convolution-capsule network (MSCCN) is constructed for crop insect pest identification. MSCCN combines the advantages of CNN, CapsNet, and multi-scale CNN to recognize various pests, including small-size ones, in complex fields. We implement a series of experiments involving the pest images in the complex fields. Experimental results with the IP102 dataset consistently produce the best identification performance with the highest accuracy and least training time. The proposed model has the advantages of good generalization, high recognition rate, and fast convergence, and provides technical support for the practical application of capsule network in crop pest identification system.

In this study, there are some problems in identifying pests in the field because the same pest may have completely different shapes and sizes during its growth period. Future research is expected to apply this method to crop pest control systems to make the system more intelligent.

**Author Contributions:** Conceptualization, C.X. and S.Z.; methodology, C.X.; software, C.Y.; writing—original draft preparation, C.X.; and writing—review and editing, X.W. 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 (Nos. 62172338 and 62072378). National Natural Science Foundation of Education Department of Shaanxi Province (No. 20JK0960).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Data were obtained from the experimental and simulation software designed in this study, which we obtained by rigorous calculation and logical reasoning.

**Acknowledgments:** We thank the project side for the use of the site and equipment required for the experiment in this study.

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

## References

1. Mercorelli, P. Denoising and harmonic detection using nonorthogonal wavelet packets in industrial applications. *J. Syst. Sci. Complex.* **2007**, *20*, 325–343. [CrossRef]
2. Mercorelli, P. A Fault Detection and Data Reconciliation Algorithm in Technical Processes with the Help of Haar Wavelets Packets. *Algorithms* **2017**, *10*, 13. [CrossRef]
3. Mercorelli, P. Biorthogonal wavelet trees in the classification of embedded signal classes for intelligent sensors using machine learning applications. *J. Frankl. Inst.* **2007**, *344*, 813–829. [CrossRef]
4. Schimmack, M.; Mercorelli, P. An Adaptive Derivative Estimator for Fault-Detection Using a Dynamic System with a Suboptimal Parameter. *Algorithms* **2019**, *12*, 101. [CrossRef]
5. Xin, J.; Buss, L.J.; Harmon, C.L.; Vergot, P.; Lester, W.J. Plant and Pest Diagnosis and Identification through DDIS. *Agric. Biol. Eng.* **2018**, *2*. [CrossRef]
6. Deng, L.; Wang, Z.; Wang, C.; He, Y.; Zhang, X. Application of agricultural insect pest detection and control map based on image processing analysis. *J. Intell. Fuzzy Syst.* **2020**, *38*, 379–389. [CrossRef]
7. Martineau, M.; Conte, D.; Raveaux, R.; Arnault, I.; Munier, D.; Venturini, G. A survey on image-based insect classification. *Pattern Recognit.* **2017**, *65*, 273–284. [CrossRef]
8. Costa, E.D.; Tjandrasa, H.; Djanali, S. Text Mining for Pest and Disease Identification on Rice Farming with Interactive Text Messaging. *Int. J. Electr. Comput. Eng.* **2018**, *8*, 1671–1683. [CrossRef]
9. Liu, X.X.; Bai, X.S.; Wang, L.H.; Ren, B.Y.; Lu, S.H. Review and Trend Analysis of Knowledge Graphs for Crop Pest and Diseases. *IEEE Access* **2019**, *7*, 62251–62264.
10. Huo, M.; Tan, J. Overview: Research Progress on Pest and Disease Identification. In *Pattern Recognition and Artificial Intelligence*; Lu, Y., Vincent, N., Yuen, P.C., Zheng, W.S., Cheriet, F., Suen, C.Y., Eds.; Springer: Cham, Switzerland, 2020. [CrossRef]
11. Li, Y.; Yang, J. Few-shot cotton pest recognition and terminal realization. *Comput. Electron. Agric.* **2020**, *169*, 105240. [CrossRef]
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
13. Ai, Y.; Sun, C.; Tie, J.; Cai, X. Research on Recognition Model of Crop Diseases and Insect Pests Based on Deep Learning in Harsh Environments. *IEEE Access* **2020**, *8*, 171686–171693. [CrossRef]
14. Xie, C.; Wang, R.; Jie, Z.; Chen, P.; Wei, D.; Rui, L.; Chen, T.; Chen, H. Multi-level learning features for automatic classification of field crop pests. *Comput. Electron. Agric.* **2018**, *152*, 233–241. [CrossRef]
15. Labaa, F.M.; Ruiz, A.; García-Sánchez, F. PestDetect: Pest Recognition Using Convolutional Neural Network. In *ICT for Agriculture and Environment*; Valencia-García, R., Alcaraz-Mármol, G., Cioppo-Morstadt, J., Vera-Lucio, N., Bucaram-Leverone, M., Eds.; Springer: Cham, Switzerland, 2019; pp. 99–108.
16. Li, Y.; Wang, H.; Dang, L.M.; Sadeghi-Niaraki, A.; Moon, H. Crop pest recognition in natural scenes using convolutional neural networks. *Comput. Electron. Agric.* **2020**, *169*, 105174. [CrossRef]
17. Li, Y.; Qian, M.; Liu, P.; Cai, Q.; Li, X.; Guo, J.; Yan, F.; Yu, F.; Yuan, K.; Yu, J. The recognition of rice images by UAV based on capsule network. *Clust. Comput.* **2019**, *22*, 9515–9524. [CrossRef]
18. Paoletti, M.E.; Haut, J.M.; Fernandez, B.R.; Plaza, J.; Plaza, A.; Li, J.; Pla, F. Capsule Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 2145–2160. [CrossRef]
19. Mensah, P.K.; Weyori, B.A.; Ayidzoe, M.A. Gabor Capsule Network for Plant Disease Detection. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 388–395.
20. Wang, D.; Xu, Q.; Xiao, Y.; Tang, J.; Luo, B. Multi-scale Convolutional Capsule Network for Hyperspectral Image Classification. *LNCS* **2019**, *11858*, 749–760.
21. Peker, M. Multi-channel capsule network ensemble for plant disease detection. *SN Appl. Sci.* **2021**, *3*, 707. [CrossRef]
22. Thenmozhi, K.; Reddy, U.S. Crop pest classification based on deep convolutional neural network and transfer learning. *Comput. Electron. Agric.* **2019**, *164*, 104906. [CrossRef]
23. Ngugi, L.C.; Abelwahab, M.; Abo-Zahhad, M. Recent advances in image processing techniques for automated leaf pest and disease recognition—A review. *Inf. Process. Agric.* **2021**, *8*, 27–51. [CrossRef]

24. Chen, H.S.; Widodo, A.M.; Wisnujati, A.; Rahaman, M.; Lin, J.C.W.; Chen, L.; Weng, C.E. AlexNet Convolutional Neural Network for Disease Detection and Classification of Tomato Leaf. *Electronics* **2022**, *11*, 951. [CrossRef]
25. Yan, P.; Su, Y.; Tian, X. Classification of Mars Lineament and Non-Lineament Structure Based on ResNet50. In Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, 25–27 August 2020; pp. 437–441. [CrossRef]
26. Wu, X.; Zhan, C.; Lai, Y.K.; Cheng, M.M.; Yang, J. IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019. [CrossRef]

*electronics*

MDPI

*Article*

# Few-Shot Learning with Collateral Location Coding and Single-Key Global Spatial Attention for Medical Image Classification

**Wenjing Shuai [1,*] and Jianzhao Li [2]**

[1] School of Electronic Engineering, Xidian University, Xi'an 710071, China
[2] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Electronic Engineering, Xidian University, Xi'an 710071, China; 19jzli@stu.xidian.edu.cn
* Correspondence: wjshuai@xidian.edu.cn; Tel.: +86-18229034065

**Abstract:** Humans are born with the ability to learn quickly by discerning objects from a few samples, to acquire new skills in a short period of time, and to make decisions based on limited prior experience and knowledge. The existing deep learning models for medical image classification often rely on a large number of labeled training samples, whereas the fast learning ability of deep neural networks has failed to develop. In addition, it requires a large amount of time and computing resource to retrain the model when the deep model encounters classes it has never seen before. However, for healthcare applications, enabling a model to generalize new clinical scenarios is of great importance. The existing image classification methods cannot explicitly use the location information of the pixel, making them insensitive to cues related only to the location. Besides, they also rely on local convolution and cannot properly utilize global information, which is essential for image classification. To alleviate these problems, we propose a collateral location coding to help the network explicitly exploit the location information of each pixel to make it easier for the network to recognize cues related to location only, and a single-key global spatial attention is designed to make the pixels at each location perceive the global spatial information in a low-cost way. Experimental results on three medical image benchmark datasets demonstrate that our proposed algorithm outperforms the state-of-the-art approaches in both effectiveness and generalization ability.

**Keywords:** few-shot learning; computational intelligence; medical image classification; spatial attention

**Citation:** Shuai, W.; Li, J. Few-Shot Learning with Collateral Location Coding and Single-Key Global Spatial Attention for Medical Image Classification. *Electronics* **2022**, *11*, 1510. https://doi.org/10.3390/electronics11091510

Academic Editor: Gemma Piella

Received: 22 April 2022
Accepted: 5 May 2022
Published: 9 May 2022

## 1. Introduction

Medicine was previously a purely artisan profession, which was highly dependent on the skills and experience of the doctors, rather than seeking to establish a standardized process for diagnosing and treating patients. On the one hand, manual analysis of large medical image datasets is a very time-consuming task [1]. On the other hand, erroneous interpretations may arise due to large smooth grayscale changes, which are imperceptible to the human eyes. Details that may be missed due to the above factors can negatively impact the treatment procedure. In recent years, the situation has begun to change because technologies such as evidence-based medicine and precision medicine have tried to inject more rigorous and data-driven methods into the this field [2].

With the increase of computing resources and data volumes, artificial intelligence has been applied in various fields, such as remote sensing image analysis [3–5], automatic driving [6–8], and privacy protection [9–11]. In the aspect of medical image analysis, deep learning has been shown to be a powerful diagnostic tool that can provide healthcare workers and patients with the exact information they need. This could give remote community health workers access to purified world medical knowledge, and it could allow physicians to greatly improve their efficiency and accuracy, while giving patients and families greater control and visibility of their healthcare. Medical image classification plays a vital role in the diagnosis process by assigning appropriate labels to certain attributes in the image. Medical image classifiers can distinguish different types of diseases in specific organs, such as breast biopsies, liver lesions, brain tissue, the lungs, and rectal cancers.

Many excellent research works have greatly advanced the field of medical image classification. Semi-supervised support vector machine was used to solve the problem of brain MRI image classification with mild cognitive impairment [12]. Peikari et al. [13] performed cluster analysis on semi-supervised learning to improve the classification performance of pathological images. In addition, several studies have explored the Generative-Adversarial-Network (GAN)-based methods, which show strong applicability in the automatic detection of retinal diseases [14], skin diseases [15], and cardiac diseases [16]. However, there are still several serious problems in the research of medical image classification. The existing deep models for medical image classification rely on a large number of labeled training samples, and their generalization performance for unseen categories is either unsatisfactory or otherwise depends on a time-consuming retraining process. Humans are very good at recognizing a new object through a very small number of samples. For example, a child only needs some pictures in a book to recognize what a "zebra" is and what a "rhinoceros" is. Inspired by the rapid learning ability of human beings, researchers seek for deep learning models to learn a new category quickly with only a small number of samples after learning a large amount of data in a certain category.

Overall, the existing deep models for medical image classification rely on a large number of labeled training samples and have poor generalization performance for unseen categories, requiring much time and computing resources to retrain. Moreover, the classification information of an image is not only related to the color of the pixel, but also to the location of the pixel, for example the location of a lesion is related to whether it is a malignant disease or not [17], while current image classification methods cannot explicitly use the location information of the pixel, making them insensitive to cues related only to the location. We propose a collateral location coding to help the network explicitly utilize the location information of each pixel to make it easier for the network to recognize cues related to location only. In addition, existing algorithms rely on local convolution and cannot properly utilize global information, which is essential for image classification. To solve this problem, we propose a single-key global spatial attention that allows each pixel in the feature map to obtain information about all features and use it as a basis for feature importance measurement.

The contributions of this paper are summarized as follows:

(1) A complete classification framework is presented for few-shot learning of medical images, which achieves excellent performance compared with the well-known few-shot learning algorithms.
(2) A collateral location coding is proposed to help the network explicitly utilize the location information.
(3) A single-key global spatial attention is designed to make the pixels at each location perceive the global spatial information in a low-cost way.
(4) Experimental results on three medical image datasets demonstrate the compelling performance of our algorithms in the few-shot task.

The remainder of this paper is structured as follows. Section 2 briefly reviews some related work in medical image classification and few-shot learning. In Section 3, our method is introduced in detail. Section 4 gives the experimental settings and the analysis of the experimental results. Finally, the conclusions and future works are described in Section 5.

## 2. Related Work

### *2.1. Medical Image Classification*

Computer-Aided Diagnosis (CAD) is an important research field, and excellent algorithms can improve the efficiency of diagnosis and reduce the chance of misdiagnosis. For example, tumors or lesions may be very small and easily missed by radiologists in the early stages, but the number of false negatives can be reduced by automatically highlighting by medical image processing.

Recently, many research works have achieved promising results in medical image classification as an important part of CAD. Annotating medical images in the real world is often time-consuming, especially when consensus is required among multiple experts. References [18,19] designed semi-supervised learning in medical image classification; the pseudo-labels were created by training a model on labeled data and then using the trained model to predict labels on unlabeled data. Furthermore, the label data and the newly generated pseudo-label data were combined as new training data. In addition, the data distribution of medical image datasets tends to be very skewed due to a large number of negative disease cases versus a small number of positive disease cases. To alleviate this problem, modified loss functions [20], cost-sensitive learning [21], oversampling or undersampling methods [22], and decision threshold shifting [23] have been designed to solve skewed class distributions.

For specific medical problems, Li et al. [24] proposed a semi-supervised graph-based algorithm to address the tongue diagnosis problem, which leverages random graph sampling techniques and label consistency modeling. De Herrera et al. [12] and Csurka et al. [25] employed semi-supervised methods to expand the training set. They first employed support vector machine (SVM) or the K-nearest neighbor (KNN) classifier trained with other multimodal (e.g., visual and textual) information to generate confidence scores for unlabeled data and then expanded the training set by manual visual retrieval. In addition, GANs were used in [16] to address the scarcity of labeled data and data domain differences in chest X-ray classification. To process high-resolution retinal fundus images for diabetic retinopathy classification, Lecouat et al. [14] proposed a patch-based classification framework and a semi-supervised GAN. Su et al. [26] proposed a local mean teacher-based self-supervised learning method that solves the kernel classification problem by enforcing local and global consistency.

### *2.2. Few-Shot Learning*

The current mainstream few-shot learning algorithms can be divided into three categories based on the data augmentation, metric learning, and meta-learning methods.

The methods based on data augmentation focus on the problem of too few samples in few-shot learning, and enhance the data themselves through a series of means, thereby transforming few-shot learning into ordinary machine learning problems. This kind of methods is mainly studied from two directions: original data enhancement and feature enhancement. The generative adversarial network proposed by Goodfellow et al. [27] employs the idea of game theory to map a certain noise distribution (generally, a Gaussian distribution) to a true distribution close to the data and realizes data enhancement from the perspective of data characteristics. On this basis, Antoniou et al. [28] proposed a data augmentation generative adversarial network to improve the quality of the model by generating data with an approximate sample distribution. Chen et al. [29] explored semantic information to design a semantic auto-encoder for higher-level data enhancement and used the image block combination method to fuse the original features of the image and the transformed features, so as to achieve the purpose of data enhancement.

The methods based on metric learning map the original data into deep features through a neural network, and the features can be used as a representation of a certain type of sample after further processing. The classification can be completed by calculating the similarity between a given sample and the representation. It usually consists of a feature embedding module, a category representation module, and a similarity measurement module. The matching network [30] employs the attention mechanism and storage memory to complete the encoding of the support set and query set samples, measures the matching degree of the two through the cosine distance, and finally, obtains the label of a given sample by the weighted average method. Moreover, for the samples that do not appear in the training, the original model does not need to be changed, and only a small amount of data can be used to complete the identification of the new category. Snell et al. [31] proposed the prototype network, which can be regarded as a general framework for deep metric learning. It represents the original data as a feature vector with feature embedding, takes the mean value of the vector of the same category as the prototype of the category, and completes the classification task by calculating the distance between the new sample and the prototype. The covariance metric network [32] takes into account the second-order features of the data, calculates the covariance to better represent the data, and achieves good performance on benchmark datasets.

Meta-learning can independently choose certain strategies to complete the learning of different tasks and study how to use previous experience to guide the existing learning, also known as "learning how to learn". Finn et al. [33] proposed a Model-Agnostic Meta-Learning (MAML) for the fast adaptation of deep networks. MAML empowers the model to independently determine the initialization of parameters with the selection of the network architecture and the optimization strategy. It obtains a global optimal value by training on the auxiliary set, which is used as the initialization value of the model on different tasks, and only needs a small number of iterations to converge on a small amount of data in a given support set. In addition, Ravi et al. [34] employed Long Short-Term Memory (LSTM) as a meta-learner to learn by taking the gradient information and the learning rate of the model as the state of the LSTM. Cheng et al. [35] proposed a meta-metric learner to integrate the matching network and LSTM.

Overall, the research on few-shot learning is still in its infancy. The breakthrough of existing algorithms in model accuracy is very dependent on deeper networks, and more emphasis is placed on experiments, which is still very much lacking in theoretical research and practical application.

## 3. Method

### 3.1. Overview

The whole dataset was divided into a training set, a validation set, and a test set, where the training set was used to train the image classifier, while the test set was further divided into support sets and query sets, where the support sets contain the few-shot labels and the query sets do not contain labels. During training, the images are first processed by the proposed collateral location coding and then fed to the feature extractor, which contains the proposed single-key global spatial attention. In the testing phase, we fixed the parameters of the feature extractor and used it to extract the image features of the support set and the query set, and finally, we used the nearest class mean for classification. The training and testing processes of our method are shown in Figures 1 and 2, respectively

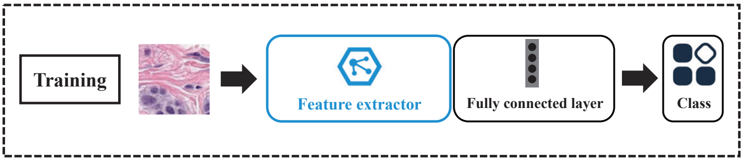


**Figure 1.** Training stage of our method. Our method follows the classical routine of training a classifier during training.

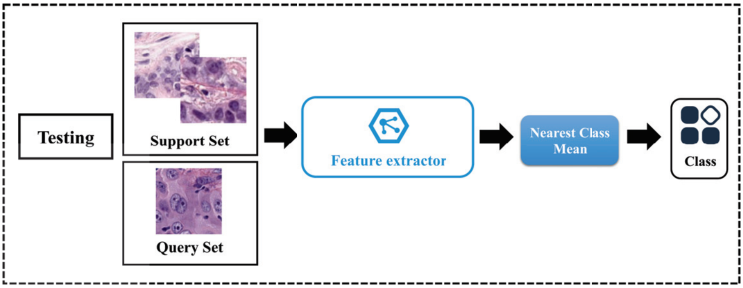


**Figure 2.** Testing stage of our method. We fix the feature extractor and use the nearest class mean method to classify the image during testing.

### *3.2. Collateral Location Coding*

When determining what kind of disease a medical image contains, the location of the lesion often has a strong correlation with the type of disease, for example the location of a lung nodule correlates with its possible development into cancer [17]. Most malignant nodules are located in the upper lobe of the lung, more commonly in the upper lobe of the right lung. Approximately two-thirds of metastases are located in the lower lobe of the lung, and approximately 60% of isolated pulmonary nodules are located in the peripheral regions of the lung. Non-calcified pulmonary nodules near the lung fissures have a lower probability of malignancy. Subpleural nodules, especially those located in the middle or lower lobe of the lung, are likely to be intrapulmonary lymph nodes. Technically, different medical images may have similarly shaped anomalies in them, but the locations of these anomalies greatly affect the classes of these medical images, so ignoring the location of the abnormalities based only on their appearance is not conducive to accurate classification. Reference [36] found that neural networks implicitly learn coarse positional information by means of padding, but existing image classification algorithms usually feed only a single RGB image into a deep neural network, which means that this process does not explicitly make use of the exact positional information of each pixel, especially considering that most classification networks end up using global pooling to eliminate spatial information, in which the average pooling will produce the same result regardless of where the key features are located.

Existing work [37] has attempted to stitch the coordinate information of the image together with the RGB image; however, the location information may be corrupted in the process due to some downsampling by the network during the convolution process; in addition, directly stitching the original coordinates is not necessarily the most helpful way for the neural network to utilize the location information, because the original coordinate information has too much difference from the color information distribution of the RGB image.

Inspired by recent advances in depth estimation [38], we propose a collateral location coding to allow the model to perceive the coordinate information of each pixel, while ensuring that the downsampling process does not corrupt the position information and allowing to reduce the difference between the position information and the distribution of RGB color information.

From any input image, we first obtain a coordinate map $p = (x, y)$ to record the position of each pixel, which is a two-channel map, recording the $x$-axis coordinate and the $y$-axis coordinate, respectively.

This coordinate map $p$ will then be coded as:

$$F_{clc}(p) = a_2 \cdot GELU(a_1 \cdot p + b_1) + b_2 \tag{1}$$

where $a_1, b_1, a_2$, and $b_2$ are linear transformation coefficients, $GELU$ is the Gaussian error linear units [39], and the linear operation of $a \cdot p + b$ can be implemented by a $1 \times 1$ convolution.

The input image will be spliced with the location feature $F_{clc}$ and then fed into the network. When the features advancing in the network encounter downsampling (e.g., pooling layer), the above process will be repeated, i.e., the features will be spliced with a location feature matching their own resolution and then sent to the next layer for processing.

### *3.3. Single-Key Global Spatial Attention*

One of the drawbacks of convolutional networks is that they can only fuse local information and each pixel can only perceive its neighbors in local spatial locations, while it is more difficult to capture remote dependencies. Self-attention is a widely adopted approach for establishing non-local connections in deep learning; yet, its huge amount of operations is still a computational burden. Inspired by recent detached attention [40], we propose a lightweight single-key global spatial attention. The process of this part is shown in Figure 3. As shown in the bottom path in the figure, the input $x$ firstly passes through a 1 × 1 convolutional layer, which does not change the number of channels, then the global pooling downsizes the spatial dimension, after another 1 × 1 convolutional layer, which does not change the number of channels, and the key of the input feature is finally obtained, i.e.,

$$K = Conv_K^{(2)}(AvgPool(Conv_K^{(1)}(x))) \tag{2}$$

The middle path in Figure 3 means that passing $x$ through a 1 × 1 convolutional layer that allows inter-channel information exchange provides the query of the input features, i.e.,

$$Q = Conv_Q(x) \tag{3}$$

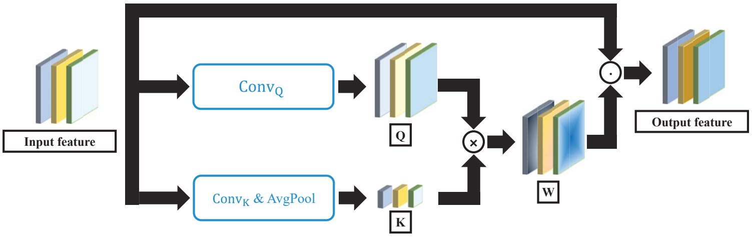


**Figure 3.** Single-key global spatial attention. We utilize a similar idea to self-attention, but the difference is that the spatial dimension of the key collapses in our approach, and each spatially located feature has to interact with only one feature instead of interacting with all features as in self-attention. We use the idea of weighting similar to SE attention [41] to weight the important features, instead of the feature generation method in self-attention.

We multiply $Q$ and $K$ and feed the result into the Sigmoid layer to obtain the weight $W$ for each spatial location, i.e.,

$$W = Q \times K^T \tag{4}$$

where T denotes the matrix transpose.

The final weighting for $x$ is accomplished by multiplying the weight matrix $W$ with the input features $x$, i.e.,

$$Out = W \cdot x \tag{5}$$

In the above process, we utilized a similar idea to self-attention, but the difference is that the spatial dimension of the key collapses in our approach, and our approach does not consume huge computational resources as self-attention does, because each spatially located feature has to interact with only one feature instead of interacting with all features as in self-attention.

In addition, we used the input features themselves as the value matrix, similar to that in self-attention; however, we did not introduce the convolution for the value, which further reduces the computational effort, and we used the idea of weighting similar to SE attention [41] to weight the important features, instead of the feature generation method in self-attention.

### *3.4. Classification*

#### 3.4.1. Training

For training, we used cross-entropy as the loss function, i.e.,

$$L(\zeta_i, \hat{\zeta}_i) = -\sum_{i=1}^{N}(\zeta_i \log \hat{\zeta}_i) \tag{6}$$

where $N$ is the number of categories, $\zeta_i$ is the ground truth distribution of the $i$-th category, and $\hat{\zeta}_i$ is the predicted distribution of the $i$-th category.

#### 3.4.2. Testing

We denote the feature extractor as $\phi$, the feature of an input image $I$ as $F_I = \phi(I)$, and $\delta_i$ as the set of the features of the $i$-th category in the support set. We used the nearest class mean to obtain a center for each category, i.e.,

$$\bar{\epsilon}_j = \frac{1}{|\delta_i|} \sum_{FI \in \delta_i} F_I \tag{7}$$

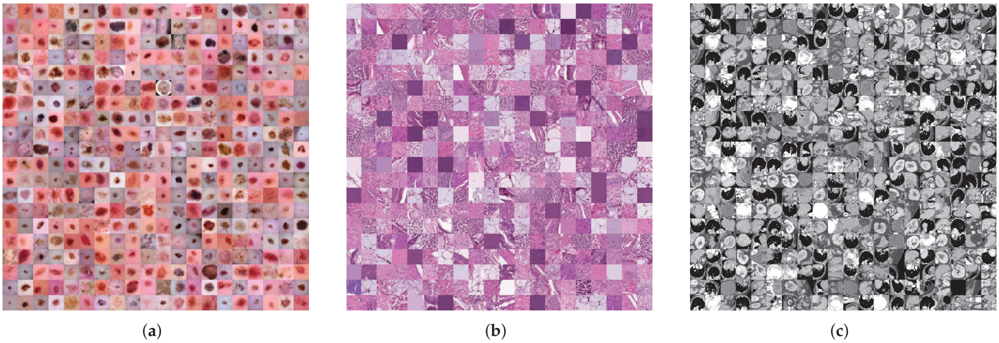
The predicted category of each sample in the query set can be obtained as:

$$Category(F_I) = \arg\min_i ||F_I - \bar{\epsilon}_i||_2 \tag{8}$$

## 4. Experimental Results and Analysis

### *4.1. Dataset Description*

The datasets employed in this paper are all from MedMNIST [42,43], which is available at https://medmnist.com/ (accessed on 31 December 2021). As a large-scale lightweight benchmark dataset for two-dimensional and three-dimensional biomedical image classification, MedMNIST has been widely used in research on medical image classification. Specifically, three datasets in MedMNIST were employed in the experiments of this paper, where the details of these datasets are presented in Figure 4 and Table 1.



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

**Figure 4.** Medical image classification datasets. (**a**) DermaMNIST. (**b**) PathMNIST. (**c**) OrganMNIST (Axial).

**Table 1.** The details of three medical image classification hldatasets.

| Datasets | Classes | Training | Validation | Test | Image Modality |
|---|---|---|---|---|---|
| DermaMNIST [44,45] | 7 | 7007 | 1003 | 2005 | Dermatoscope |
| PathMNIST [46] | 9 | 89,996 | 10,004 | 7180 | Pathology |
| OrganMNIST (Axial) [47,48] | 11 | 34,581 | 6491 | 17,778 | Abdominal CT |

**DermaMNIST** is based on HAM10000 [44,45], which is a collection of multi-source dermoscopic images of large common pigmented skin lesions. The source images with 3 × 600 × 450 pixels were resized to 3 × 28 × 28 pixels. The dataset consists of 10,015 dermoscopic images, which are divided into seven different diseases to form a multi-class classification task. The images were divided into a training set, verification set, and test set in the ratio of 7:1:2.

**PathMNIST** is based on a prior study [46] and is mainly used to predict survival in colorectal cancer histological sections. The source images with 3 × 224 × 224 pixels were resized to 3 × 28 × 28 pixels. In [46], a dataset (NCT-CRC-HE-100K) containing 100,000 non-overlapping image patches from hematoxylin- and eosin-stained histological images were split as 9:1 into a training set and verification set. In addition, a dataset (CRC-VAL-HE-7K) with 7180 image patches from different clinical centers was treated as the test set. The PathMNIST dataset consists of nine types of organizations, which allows for multiple classification tasks.

**OrganMNIST(Axial)** is the axial acquisition from 3D Computed Tomography (CT) in the Liver Tumor Segmentation Benchmark (LiTS) [47]. The organ labels in OrganMNIST (Axial) were obtained from boundary box annotations of 11 body organs in another study [48]. The original image was resized to 1 × 28 × 28 pixels, which classifies 11 body organs into multiple categories. In detail, the training and validation set were selected from 115 and 16 CT scans in the source training set, respectively. The test set was constructed with 70 CT scans from the source test set.

### *4.2. Experimental Setup*

For the sake of fairness, all experiments in this paper were implemented on the PyTorch framework in an NVIDIA GeForce RTX 3090. In the practical implementation, we randomly selected three categories as the training set, two categories as the validation set, and the remaining two categories as the test set in DermaMNIST, so 2-way 1-shot and 2-way 5-shot were performed in the comparative experiments. For PathMNIST, we randomly selected three categories as the training set, three categories as the validation set, and the remaining three categories as the test set. For OrganMNIST, we randomly selected five categories as the training set, three categories as the validation set, and the remaining three categories as the test set. In addition, 3-way 1-shot and 3-way 5-shot were performed in the PathMNIST and OrganMNIST. We used ResNet18 [49] as the backbone, where we added the proposed single-key global spatial attention module at the end of each convolution block. We adopted the optimizer of SGD with a momentum of 0.9. The learning rate was 0.1. We also report the 95% confidence interval, and the performances were averaged over 1000 generated classification tasks.

### *4.3. Comparing with State-of-the-Art Algorithms*

In order to quantify the superiority of our proposed algorithm, five well-known few-shot learning algorithms were selected as the comparison algorithms, including the MatchingNet [30], MAML [33], Prototype Net [31], Relation Net [50], and Transductive Propagation Network (TPN) [51].

For DermaMNIST, as can be seen from the experimental results in Table 2, our method achieved the best results on 2-way 1-shot and 5-shot. Specifically, our method outperformed the state-of-the-art method by 3.25% and 1.86% in 1-shot and 5-shot, respectively. We also show the loss curve and the validation accuracy curve of the proposed method on the DermaMNIST dataset, in Figures 5 and 6, respectively.

Table 3 shows the experimental results on the PathMNIST dataset; our method outperformed all existing methods. The results on the OrganMNIST dataset are shown in Table 4; our method achieved the best performance with the highest accuracy and the lowest confidence interval.

**Table 2.** The accuracy comparison of different methods on the DermaMNIST dataset.

| **Method** | **2-Way** | |
|---|---|---|
| | **1-Shot** | **5-Shot** |
| MatchingNet | 55.52 ± 1.14% | 61.91 ± 1.57% |
| MAML | 56.14 ± 0.97% | 63.27 ± 1.12% |
| PrototypeNet | 56.84 ± 0.88% | 62.74 ± 1.18% |
| Relation Net | 58.74 ± 0.84% | 63.82 ± 1.20% |
| TPN | 60.12 ± 0.86% | 67.52 ± 1.14% |
| **Ours** | **63.37 ± 0.80%** | **69.38 ± 1.03%** |

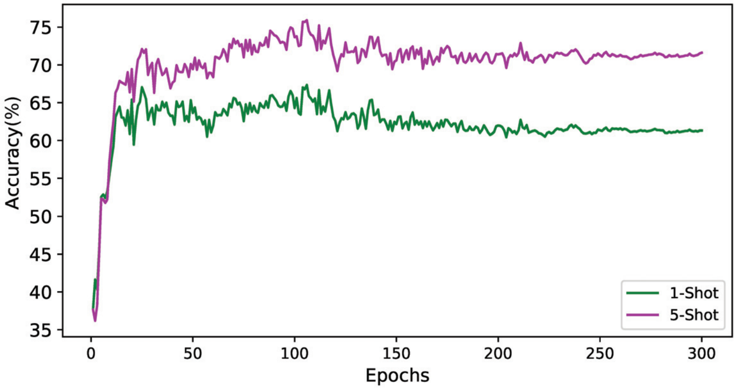


**Figure 5.** Validation accuracy curve on the DermaMNIST dataset.

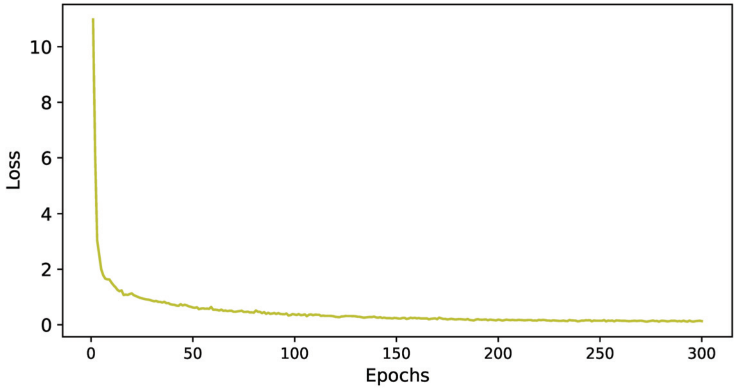


**Figure 6.** Loss curve on the DermaMNIST dataset.

**Table 3.** The accuracy comparison of different methods on the PathMNIST dataset.

| **Method** | **3-Way** | |
|---|---|---|
| | **1-Shot** | **5-Shot** |
| MatchingNet | 46.38 ± 0.82% | 53.28 ± 1.29% |
| MAML | 51.58 ± 0.81% | 58.39 ± 0.92% |
| PrototypeNet | 51.29 ± 0.77% | 59.19 ± 0.82% |
| Relation Net | 53.48 ± 0.81% | 60.73 ± 0.87% |
| TPN | 52.91 ± 0.83% | 59.29 ± 0.84% |
| **Ours** | **54.82** ± **0.78%** | **61.92** ± **0.81%** |

**Table 4.** The accuracy comparison of different methods on the OrganMNIST dataset.

| **Method** | **3-Way** | |
|---|---|---|
| | **1-Shot** | **5-Shot** |
| MatchingNet | 44.59 ± 0.96% | 50.84 ± 1.12% |
| MAML | 48.47 ± 0.87% | 56.86 ± 0.96% |
| PrototypeNet | 49.39 ± 0.83% | 57.83 ± 0.72% |
| Relation Net | 50.93 ± 0.84% | 58.61 ± 0.89% |
| TPN | 51.86 ± 0.87% | 57.35 ± 0.85% |
| **Ours** | **53.48** ± **0.81%** | **59.38** ± **0.84%** |

### *4.4. Ablation Experiments*

In this subsection, the ablation experiments are performed to demonstrate the effectiveness of our innovation. The ablation results on DermaMNIST, PathMNIST, and OrganMNIST are shown in Tables 5–7, respectively. Both of the proposed contributions improved the performance because collateral-type location coding allows the model to exploit feature information related to location only, while single-key global spatial attention allows the model to make each pixel in the feature map perceive global information in a cost-effective manner.

**Table 5.** Ablation on the DermaMNIST dataset.

| **Method** | **2-Way** | |
|---|---|---|
| | **1-Shot** | **5-Shot** |
| Baseline | 59.28 ± 1.01% | 63.81 ± 1.29% |
| + Collateral Location Coding | 61.27 ± 0.98% | 65.72 ± 1.21% |
| + Single-Key Global Spatial Attention | 62.79 ± 0.91% | 65.14 ± 1.18% |
| **Full** | **63.37** ± **0.80%** | **69.38** ± **1.03%** |

**Table 6.** Ablation on the PathMNIST dataset.

| **Method** | **3-Way** | |
|---|---|---|
| | **1-Shot** | **5-Shot** |
| Baseline | 49.91 ± 0.95% | 56.28 ± 1.04% |
| + Collateral Location Coding | 51.39 ± 0.91% | 58.21 ± 0.97% |
| + Single-Key Global Spatial Attention | 52.96 ± 0.84% | 58.49 ± 0.89% |
| **Full** | **54.82** ± **0.78%** | **61.92** ± **0.81%** |

**Table 7.** Ablation on the OrganMNIST dataset.

| **Method** | **3-Way** | |
|---|---|---|
| | **1-Shot** | **5-Shot** |
| Baseline | 50.48 ± 0.98% | 55.71 ± 1.07% |
| + Collateral Location Coding | 51.41 ± 0.93% | 57.39 ± 0.91% |
| + Single-Key Global Spatial Attention | 51.83 ± 0.88% | 57.57 ± 0.89% |
| **Full** | **53.48** ± **0.81%** | **59.38** ± **0.84%** |

## 5. Conclusions

In this paper, we proposed a few-shot learning framework for medical image classification, in which we specifically proposed a collateral location encoding to help the network recognize only location-dependent features, and we proposed a single-key global spatial attention that allows the model to perceive global spatial information in a cost-effective manner. Experiments on three publicly available medical datasets confirmed the effectiveness of our algorithm. Noticing that a large amount of valuable medical data is underused, we find it urgent to fuse various medical classification data sources seeking a further boost in performance. Therefore, in our future work, we will focus on how to embed unannotated samples from different medical data sources into a few-shot learning framework to further improve model effectiveness.

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

**Funding:** This work was supported by the National Natural Science Foundation of China (61906148).

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

## 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**, 1–14. [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. Gong, M.; Liang, Y.; Shi, J.; Ma, W.; Ma, J. Fuzzy c-means clustering with local information and kernel metric for image segmentation. *IEEE Trans. Image Process.* **2012**, *22*, 573–584. [CrossRef] [PubMed]
5. Gong, M.; Zhou, Z.; Ma, J. Change detection in synthetic aperture radar images based on image fusion and fuzzy clustering. *IEEE Trans. Image Process.* **2011**, *21*, 2141–2151. [CrossRef]
6. Gong, M.; Feng, K.y.; Fei, X.; Qin, A.K.; Li, H.; Wu, Y. An Automatically Layer-wise Searching Strategy for Channel Pruning Based on Task-driven Sparsity Optimization. *IEEE Trans. Circ. Syst. Video Technol.* **2022**, 1. [CrossRef]
7. Wu, Y.; Liu, J.W.; Zhu, C.Z.; Bai, Z.F.; Miao, Q.G.; Ma, W.P.; Gong, M.G. Computational intelligence in remote sensing image registration: A survey. *Int. J. Autom. Comput.* **2021**, *18*, 1–17. [CrossRef]
8. 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]
9. Wu, Y.; Xiao, Z.; Liu, S.; Miao, Q.; Ma, W.; Gong, M.; Xie, F.; Zhang, Y. A Two-Step Method for Remote Sensing Images Registration Based on Local and Global Constraints. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 5194–5206. [CrossRef]
10. Li, H.; Li, J.; Zhao, Y.; Gong, M.; Zhang, Y.; Liu, T. Cost-Sensitive Self-Paced Learning With Adaptive Regularization for Classification of Image Time Series. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 11713–11727. [CrossRef]
11. Wang, Z.; Li, J.; Liu, Y.; Xie, F.; Li, P. An Adaptive Surrogate-Assisted Endmember Extraction Framework Based on Intelligent Optimization Algorithms for Hyperspectral Remote Sensing Images. *Remote Sens.* **2022**, *14*, 892. [CrossRef]
12. García Seco de Herrera, A.; Markonis, D.; Joyseeree, R.; Schaer, R.; Foncubierta-Rodríguez, A.; Müller, H. Semi–supervised learning for image modality classification. In *International Workshop on Multimodal Retrieval in the Medical Domain*; Springer: Berlin, Germany, 2015.
13. Peikari, M.; Salama, S.; Nofech-Mozes, S.; Martel, A.L. A cluster-then-label semi-supervised learning approach for pathology image classification. *Sci. Rep.* **2018**, *8*, 7193. [CrossRef] [PubMed]

14. Lecouat, B.; Chang, K.; Foo, C.S.; Unnikrishnan, B.; Brown, J.M.; Zenati, H.; Beers, A.; Chandrasekhar, V.; Kalpathy-Cramer, J.; Krishnaswamy, P. Semi-supervised deep learning for abnormality classification in retinal images. *arXiv* **2018**, arXiv:1812.07832.
15. Springenberg, J.T. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv* **2015**, arXiv:1511.06390.
16. Madani, A.; Moradi, M.; Karargyris, A.; Syeda-Mahmood, T. Semi-supervised learning with generative adversarial networks for chest X-ray classification with ability of data domain adaptation. In Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018; pp. 1038–1042.
17. Armato, S.G.; Li, F.; Giger, M.L.; MacMahon, H.; Sone, S.; Doi, K. Lung cancer: Performance of automated lung nodule detection applied to cancers missed in a CT screening program. *Radiology* **2002**, *225*, 685–692. [CrossRef] [PubMed]
18. Liu, Q.; Yu, L.; Luo, L.; Dou, Q.; Heng, P.A. Semi-Supervised Medical Image Classification With Relation-Driven Self-Ensembling Model. *IEEE Trans. Med. Imaging* **2020**, *39*, 3429–3440. [CrossRef]
19. Gyawali, P.K.; Ghimire, S.; Bajracharya, P.; Li, Z.; Wang, L. Semi-supervised medical image classification with global latent mixing. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2020; pp. 604–613.
20. Phan, H.; Krawczyk-Becker, M.; Gerkmann, T.; Mertins, A. DNN and CNN with weighted and multi-task loss functions for audio event detection. *arXiv* **2017**, arXiv:1708.03211.
21. Khan, S.H.; Hayat, M.; Bennamoun, M.; Sohel, F.A.; Togneri, R. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 3573–3587.
22. Han, W.; Huang, Z.; Li, S.; Jia, Y. Distribution-sensitive unbalanced data oversampling method for medical diagnosis. *J. Med Syst.* **2019**, *43*, 1–10. [CrossRef]
23. Yu, H.; Sun, C.; Yang, X.; Yang, W.; Shen, J.; Qi, Y. ODOC-ELM: Optimal decision outputs compensation-based extreme learning machine for classifying imbalanced data. *Knowl. Based Syst.* **2016**, *92*, 55–70. [CrossRef]
24. Li, C.H.; Yuen, P.C. Semi-supervised learning in medical image database. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin, Germany, 2001; pp. 154–160.
25. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 702–703.
26. Su, H.; Shi, X.; Cai, J.; Yang, L. Local and global consistency regularized mean teacher for semi-supervised nuclei classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2019; pp. 559–567.
27. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the 28th Conference on Neural Information Processing Systems (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014; p. 27.
28. Antoniou, A.; Storkey, A.; Edwards, H. Data augmentation generative adversarial networks. *arXiv* **2017**, arXiv:1711.04340.
29. Chen, Z.; Fu, Y.; Chen, K.; Jiang, Y.G. Image block augmentation for one-shot learning. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 3379–3386. [CrossRef]
30. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; p. 29.
31. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; p. 30.
32. Li, W.; Xu, J.; Huo, J.; Wang, L.; Gao, Y.; Luo, J. Distribution consistency based covariance metric networks for few-shot learning. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 8642–8649. [CrossRef]
33. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
34. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1–11.
35. Cheng, Y.; Yu, M.; Guo, X.; Zhou, B. Few-shot learning with meta metric learners. *arXiv* **2019**, arXiv:1901.09890.
36. Islam, M.A.; Jia, S.; Bruce, N.D. How much position information do convolutional neural networks encode? *arXiv* **2020**, arXiv:2001.08248.
37. Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; Li, L. Solo: Segmenting objects by locations. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2020; pp. 649–665.
38. Gonzalez, J.L.; Kim, M. PLADE-Net: Towards Pixel-Level Accuracy for Self-Supervised Single-View Depth Estimation with Neural Positional Encoding and Distilled Matting Loss. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 6851–6860.
39. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
40. Liu, H.; Liu, F.; Fan, X.; Huang, D. Polarized self-attention: Towards high-quality pixel-wise regression. *arXiv* **2021**, arXiv:2107.00782.
41. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.

42. Yang, J.; Shi, R.; Ni, B. MedMNIST Classification Decathlon: A Lightweight AutoML Benchmark for Medical Image Analysis. In Proceedings of the 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), Nice, France, 13–16 April 2021; pp. 191–195.
43. Yang, J.; Shi, R.; Wei, D.; Liu, Z.; Zhao, L.; Ke, B.; Pfister, H.; Ni, B. MedMNIST v2: A Large-Scale Lightweight Benchmark for 2D and 3D Biomedical Image Classification. *arXiv* **2021**, arXiv:2110.14795.
44. Tschandl, P.; Rosendahl, C.; Kittler, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Sci. Data* **2018**, *5*, 180161. [CrossRef]
45. Codella, N.; Rotemberg, V.; Tschandl, P.; Celebi, M.E.; Dusza, S.; Gutman, D.; Helba, B.; Kalloo, A.; Liopyris, K.; Marchetti, M.; et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv* **2019**, arXiv:1902.03368.
46. Kather, J.N.; Krisam, J.; Charoentong, P.; Luedde, T.; Herpel, E.; Weis, C.A.; Gaiser, T.; Marx, A.; Valous, N.A.; Ferber, D.; et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS Med.* **2019**, *16*, e1002730. [CrossRef] [PubMed]
47. Bilic, P.; Christ, P.F.; Vorontsov, E.; Chlebus, G.; Chen, H.; Dou, Q.; Fu, C.W.; Han, X.; Heng, P.A.; Hesser, J.; et al. The liver tumor segmentation benchmark (lits). *arXiv* **2019**, arXiv:1901.04056.
48. Xu, X.; Zhou, F.; Liu, B.; Fu, D.; Bai, X. Efficient Multiple Organ Localization in CT Image Using 3D Region Proposal Network. *IEEE Trans. Med. Imaging* **2019**, *38*, 1885–1898. [CrossRef] [PubMed]
49. 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.
50. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1199–1208.
51. Liu, Y.; Lee, J.; Park, M.; Kim, S.; Yang, E.; Hwang, S.J.; Yang, Y. Learning to propagate labels: Transductive propagation network for few-shot learning. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019; pp. 1–14.

*electronics*

*Article*

# Evolutionary Optimization Based Set Joint Integrated Probabilistic Data Association Filter

**Shuang Liang [1], Yun Zhu [2,*] and Hao Li [3,*]**

[1] Academy of Advanced Interdisciplinary Research, Xidian University, Xi'an 710071, China; sliang@xidian.edu.cn
[2] School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
[3] School of Electronic Engineering, Xidian University, Xi'an 710071, China
[*] Correspondence: yunzhu@snnu.edu.cn (Y.Z.); haoli@xidian.edu.cn (H.L.); Tel.: +86-159-9167-1149 (Y.Z.); +86-029-8820-2236 (H.L.)

**Abstract:** The joint integrated probabilistic data association (JIPDA) algorithm is widely used for the automatic tracking of multiple targets, but it has the well-known problem of track coalescence. By optimizing the posterior density, the accuracy of the target state estimation can be improved. Motivated by this idea, we developed a novel evolutionary optimization based joint integrated probabilistic data association (EOJIPDA) filter to overcome the coalescence problem of the JIPDA filter. The trace for the covariance matrix of the posterior density is used as the objective function for the above optimization problem. It is shown that the accuracy of the target state estimation can be improved by reducing the trace. Evolutionary optimization was employed to minimize the trace and optimize the posterior density. More specifically, we enumerated all the possible permutations of the targets and assign a unique index to each permutation. The resulting indices were randomly assigned to all possible association hypothesis events. Each assignment indicated one possible gene in the evolutionary algorithm. This process was repeated several times to arrive at the initial population. An illustrative example shows that the EOJIPDA filter can effectively improve the accuracy of state estimation. Numerical studies are presented for two challenging multi-target tracking scenarios with clutter and missed detections. The experimental results demonstrate that the EOJIPDA filter provides better tracking accuracy than traditional coalescence-avoiding methods.

**Keywords:** multi-target tracking; evolutionary optimization; random finite set; joint integrated probabilistic data association

**Citation:** Liang, S.; Zhu, Y.; Li, H. Evolutionary Optimization Based Set Joint Integrated Probabilistic Data Association Filter. *Electronics* **2022**, *11*, 582. https://doi.org/10.3390/electronics11040582

Academic Editor: Marco Mussetta

Received: 10 January 2022
Accepted: 10 February 2022
Published: 15 February 2022

## 1. Introduction

Multi-target tracking (MTT) is one of the most important low-level techniques used in radar, computer vision, Internet of things [1,2], and other surveillance systems [1–5]. Traditional MTT methods usually regard the MTT problem as the tracking of multiple single-targets. Examples of such methods are the multiple hypothesis tracking (MHT) [6,7] and joint probabilistic data association (JPDA) [8] filters. The essence of these filters is the association between the measurements and the targets being tracked. The MHT filter propagates all possible association hypothesis events over time. The target states can be estimated accurately from sufficient events. However, the main drawback of the MHT filter is its heavy computation. Compared with the MHT filter, the JPDA filter requires less computation and is effective at MTT. Nevertheless, the JPDA filter has the track coalescence problem, whereby tracks tend to coalesce when the targets are closely spaced. The JPDA filter has a well-known drawback in that it assumes that the number of targets and their initial states are known a priori. Furthermore, the number of targets remains constant during tracking.

In practice, the number and states of targets are usually unknown in advance and the number of targets may vary with time. The automatic tracking system is a natural choice to track targets in this situation. Both target and clutter measurements are used by the automatic tracking system to initiate tracks. To distinguish between false and true tracks, it is necessary to measure the quality of each track. The joint integrated probabilistic data association (JIPDA) filter uses the probability of target existence to measure the quality of the track and is effective for automatic tracking [9]. However, the JIPDA filter also suffers from the track coalescence problem.

To overcome the track coalescence problem of the JIPDA filter and further improve the tracking accuracy, we combine the advantages of the RFS theory and the standard JIPDA filter and propose a novel filter, named evolutionary optimization based joint integrated probabilistic data association (EOJIPDA). The essential reason for the track coalescence is that the overlap of the tracking gates leads to the association uncertainty problem. In this case, the posterior density becomes multimodal and the estimation of the multi-target state becomes less accurate. The trace of the covariance matrix is a measure of the multimodality of the posterior density [10,11]. To improve the accuracy of the multi-target state estimation, we used the evolutionary computation approach to minimize the trace of the covariance matrix and optimize the posterior density. First, it is necessary to define a population of possible solutions in the search space for the optimization problem. We enumerated all the possible permutations of the targets and assigned a unique index to each permutation. The resulting indices were randomly assigned to all the possible data association events. Each assignment indicated one possible gene in the evolutionary algorithm. This process was repeated several times until we obtained the initial population. Each candidate solution was assigned a fitness value based on the cost function. Fitter individuals have a higher chance of mating, producing more "fitter" individuals. The crossover and mutation are then applied to the population to generate the new offspring. This process is repeated until the stopping criterion is reached. The main contributions of the proposed EOJIPDA filter are summarized as follows.

(1) Although the JIPDA filter is effective for automatic MTT, few improvements have been proposed to overcome its coalescence problem. We propose to optimize the posterior density of the JIPDA filter using the RFS theory, when target identity is irrelevant.

(2) To the best of our knowledge, this is the first work to use the computational intelligence technique to improve the performance of the data association based filter. We model the optimization of the posterior density as an evolutionary optimization problem, improving the accuracy of the state estimation.

(3) The illustrative example shows that the EOJIPDA filter can effectively improve the accuracy of the state estimation. Simulation results obtained from two challenging MTT scenarios demonstrate the effectiveness of the EOJIPDA filter in terms of the optimal sub-pattern assignment (OSPA) multi-target miss distance.

The organization of this paper is as follows. Section 2 introduces the target and measurement models, and the necessary background on the JIPDA filter. In Section 3, the proposed EOJIPDA filter is described in detail. Section 4 evaluates the performance of the EOJIPDA filter using two different tracking scenarios. Finally, Section 5 presents concluding remarks.

## 2. Related Work

To date, several approaches have been proposed to overcome track coalescence, such as the exact nearest-neighbor JPDA (ENNJPDA) filter [12], the JPDA* filter [13], the set JPDA (SJPDA) filter [10], the Kullback–Leibler SJPDA (KLSJPDA) filter [10], the multi-objective JPDA (MOJPDA) [14], and the nearest-neighbor SJPDA (NNSJPDA) filter [15]. Motivated by the JPDA* filter [13], the JIPDA* filter [16] has also been proposed to overcome the track coalescence problem of the traditional JIPDA filter. In the JIPDA* filter, the target states are estimated by selecting one association hypothesis event and the other association hypothesis events are pruned. The track coalescence problem can be avoided effectively by the JIPDA* filter, but the association hypothesis events being pruned may contain some useful information. Our recent work [11] proposes to use the information contained in all the association hypothesis events by optimizing the posterior density. The proposed algorithm strongly depends on initializations and improper initializations may result in bad local minima.

Finite set statistics (FISST)-based MTT algorithms [17], such as the probability hypothesis density (PHD) filter [18], the cardinalized PHD (CPHD) filter [19], the multi-Bernoulli (MB) filter [20], the label multi-Bernoulli (LMB) filter [21], and the generalized label multi-Bernoulli (GLMB) filter [22,23], have attracted significant attention in recent years. These filters use the random finite set (RFS) theory to model the uncertainty in systems. Among them, the PHD and CPHD filters are the moment approximations of the Bayes multi-target filter and have no analytical expression for the posterior multi-target density. Unlike the PHD and CPHD filters, the MB filter propagates parameters of the multi-Bernoulli RFS instead of the posterior multi-target density. By using the labeled RFS, the LMB and GLMB filters have also been proposed as approximations of the multi-target Bayes filter. These filters can be implemented by the Gaussian mixture (GM) method or the sequential Monte Carlo (SMC) method, depending on the characteristics of the tracking system. In Table 1, we list all the above-mentioned existing MTT methods.

**Table 1.** Existing MTT methods.

| | **JPDA-Based Multi-Target Tracking** | |
|---|---|---|
| **Reference** | **Filter** | **Characteristic** |
| [8] | JPDA | The number of targets and their initial states are known a priori.<br>The number of targets remains constant during tracking. |
| [12] | ENNJPDA | After a measurement-to-target assignment is performed, tracks are only updated by a single measurement. |
| [13] | JPDA* | The best data association hypothesis is chosen to calculate the measurement-to-target probabilities. |
| [10] | SJIPDA | When all hypotheses are combined together for full optimization, the computation burden can be huge. |
| [10] | KLSJPDA | The optimal Gaussian approximations are provided in the Kullback–Leibler sense. |
| [14] | MOJPDA | The cost function is a linear combination of multiple objective functions. |
| [15] | NNSJPDA | A pair selection criterion is used for the iterative optimization. |
| | **JIPDA-Based Multi-Target Tracking** | |
| **Reference** | **Filter** | **Characteristic** |
| [9] | JIPDA | The number of targets can be unknown and time-varying.<br>Measurements are used to initiate tracks. |
| [16] | JIPDA* | It combines the JIPDA filter with the JPDA* scheme.<br>The hypothesis events being pruned may contain some useful information. |
| [11] | CCJIPDA | It strongly depends on initializations and improper initializations may result in bad local minima. |

**Table 1.** *Cont.*

| RFS Based Algorithms | | |
|---|---|---|
| **Reference** | **Filter** | **Characteristic** |
| [18] | PHD | Moment approximation of the Bayes multi-target filter.<br>It has no analytical expression for the posterior multi-target density. |
| [19] | CPHD | Besides the moment approximation, the filter also propogates the cardinality distribution. |
| [20] | MB | A set of multi-Bernoulli parameters are used to characterize the posterior multi-target RFS. |
| [21] | LMB | A generalization of the multi-Bernoulli filter. |
| [22,23] | GLMB | The GLMB density is closed under the multi-target prediction and update operations. |

JPDA* denotes a coalescence-avoiding version of JPDA, as shown in [13].

## 3. Background

### 3.1. Target and Measurement Assumptions

We consider the nearly constant velocity model in this paper. Assuming that the target state at time $k$ is denoted as $\boldsymbol{x}_k$, the target motion is modelled as follows

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{u}_k \tag{1}$$

where $\boldsymbol{F}_k$ is the transition matrix and $\boldsymbol{u}_k \sim \mathcal{N}(0, \boldsymbol{Q}_k)$ is zero-mean Gaussian distributed process noise with covariance $\boldsymbol{Q}_k$. If the number of all potential targets (tracks) at time $k$ is $n_k$, the augmented vector of labeled target states is

$$\boldsymbol{X}_k = \left[ \left(\boldsymbol{x}_k^1\right)^{\mathrm{T}}, \left(\boldsymbol{x}_k^2\right)^{\mathrm{T}}, \ldots, \left(\boldsymbol{x}_k^{n_k}\right)^{\mathrm{T}} \right]^{\mathrm{T}} \tag{2}$$

At time $k$, the relation between the target $\boldsymbol{x}_k$ and its measurement $\boldsymbol{z}_k$ is described as

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{w}_k \tag{3}$$

where $\boldsymbol{H}_k$ is the observation matrix and $\boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{R}_k)$ denotes that $\boldsymbol{w}_k$ follows the Gaussian distribution with zero-mean and covariance $\boldsymbol{R}_k$. The target-originated measurements are detected with the probability of detection $P_D$. Measurements may also originate from clutters. It is assumed that the number of clutter measurements follows the Poisson distribution and the average number of clutter measurements at each time step is $r = \lambda|\mathrm{FoV}|$, where $\lambda$ is the intensity of clutter and $|\mathrm{FoV}|$ denotes the sensor's field-of-view. The set of the validated measurements received at time $k$ is given by

$$\boldsymbol{Z}_k = \left\{ \boldsymbol{z}_k^1, \boldsymbol{z}_k^2, \cdots, \boldsymbol{z}_k^{m_k} \right\} \tag{4}$$

where $m_k$ denotes the number of measurements. The sequence of measurement sets accumulated up to time $k$ is described as $\boldsymbol{Z}_{1:k} = \{\boldsymbol{Z}_1, \boldsymbol{Z}_2, \cdots, \boldsymbol{Z}_k\}$.

### 3.2. Joint Integrated Probabilistic Data Association Filter

In the JIPDA filter, association hypothesis events are formed by assigning the validated measurements to the targets being tracked. From this, the target state $\boldsymbol{x}_k^t$ and covariance $\boldsymbol{P}_k^t$ are estimated for each track $t$ at time $k$. What's more, the estimation of the probability of target existence $r_k^t$ is also taken into considered. Therefore, a track $t$ can be completely described by the parameter set $\{r_k^t, \mathcal{N}(\boldsymbol{x}_k^t, \boldsymbol{P}_k^t))\}$, where $\mathcal{N}(\boldsymbol{x}_k^t, \boldsymbol{P}_k^t)$ is a Gaussian PDF with mean $\boldsymbol{x}_k^t$ and covariance $\boldsymbol{P}_k^t$. At time $k$ + 1, the main steps used to estimate the parameter set are as follows.

- Step 1: Predicting the target state $\boldsymbol{x}^t_{k+1|k}$, the covariance $\boldsymbol{P}^t_{k+1|k}$, and the probability of target existence $r^t_{k+1|k}$ for each track $t$.

The predicted target state $\boldsymbol{x}^t_{k|k-1}$ and covariance $\boldsymbol{P}^t_{k|k-1}$ are estimated using the Kalman filter. The propagation of the target' existence probability follows the Markov chain one,

$$r^t_{k+1|k} = p_{11} r^t_k + p_{21}(1 - r^t_k) \tag{5}$$

where $p_{11}$ and $p_{21}$ are the Markov chain coefficients.

- Step 2: The tracking gate is generated for each target to select the validated measurements.

After the prediction, gating is performed for each track by defining an area around the predicted track. The area is named as "gate" and only the measurements falling within the gate are regarded as the validated measurements and are used to update the track. The authors of [24] present various gating techniques, among which the ellipsoidal gate [25] is widely used. The ellipsoidal gating area for track $t$ at time $k$ is defined as follows,

$$\mathcal{R}^t_k = \left\{ \boldsymbol{z}^j_k \in \mathbb{R}^{d_z} \middle| d^t(\boldsymbol{z}^j_k) \leq G \right\} \tag{6}$$

where

$$d^t(\boldsymbol{z}^j_k) = v(k,t,j)^{\mathrm{T}} \left(\boldsymbol{S}^t_k\right)^{-1} v(k,t,j)^{\mathrm{T}} \tag{7}$$

denotes the distance between predicted measurement $\boldsymbol{z}^t_{k|k-1}$ and received measurement $\boldsymbol{z}^j_k$, $v(k,t,j) = \boldsymbol{z}^j_k - \boldsymbol{z}^t_{k|k-1}$ denotes the difference between $\boldsymbol{z}^t_{k|k-1}$ and $\boldsymbol{z}^j_k$, $\boldsymbol{S}^t_k$ is the innovation covariance, $d_z$ is the measurement dimension, and $G$ is the threshold that leads to a specified gating probability $P_W$.

- Step 3: Association hypothesis events are formed by associating the validated measurements with the tracks.

The posterior probability of the association hypothesis event $\theta_h$ is computed as

$$P(\theta_h) = C^{-1} \prod_{t \in T^h_0} \left(1 - P_D P_W r^t_{k-1}\right) \times \prod_{t \in T^h_1} \left( P_D P_W r^t_{k-1} \frac{p^t_h V_k}{\hat{m}_k} \right) \tag{8}$$

where $C$ is the normalization constant; $p^t_h = f^t(\boldsymbol{z}^t_k | \boldsymbol{Z}_{1:k-1}) / P_W$ with $P_W$ denoting the gating probability and $\boldsymbol{Z}_{1:k-1}$ is the set of measurements accumulated up to time $k-1$; $P_D$ is the probability of detection; $V_k$ is the cluster volume; $\hat{m}_k$ is the number of clutter measurement; $T^h_0$ denotes the set of tracks associated with no measurement; and $T^h_1$ denotes the set of tracks associated with one measurement.

Using the posterior probability $P(\theta_h)$, the target existence probability of track $t$ in $\theta_h$ is computed as

$$r^{t|h}_k = \left(1 - \sum_{i=1}^{m_k} \xi^{t|h}_i \right) r^{t|h}_{k,0} + \sum_{i=1}^{m_k} \xi^{t|h}_i r^{t|h}_{k,i} \tag{9}$$

where $r^{t|h}_{k,0}$ denotes the probability that track $t$ is not associated with any measurement, $r^{t|h}_{k,i}$ denotes the probability that track $t$ is associated with measurement $\boldsymbol{z}^i_k$ in $\theta_h$, and the parameter $\xi^{t|h}_i$ indicates whether the track $t$ is associated with a measurement, i.e.,

$$\begin{aligned} \xi^{t|h}_i &= 1\,, \text{ if track } t \text{ is associated with measuremt } \boldsymbol{z}^i_k \\ \xi^{t|h}_i &= 0\,, \text{ otherwise} \end{aligned} \tag{10}$$

The probabilities $r_{k,0}^{t|h}$ and $r_{k,i}^{t|h}$ are computed as follows

$$r_{k,0}^{t|h} = \frac{(1 - P_D^t P_W^t) r_{k|k-1}^t}{1 - P_D^t P_W^t r_{k|k-1}^t} P(\theta_h),\ r_{k,i}^{t|h} = P(\theta_h) \tag{11}$$

Using the posterior probability $P(\theta_h)$ and the target existence probability $r_k^{t|h}$, the target state and the error covariance of track $t$ in $\theta_h$ are estimated as

$$\boldsymbol{x}_k^{t|h} = \left(1 - \sum_{i=1}^{m_k} \xi_i^{t|h}\right) \boldsymbol{x}_{k|k-1}^t + \sum_{i=1}^{m_k} \xi_i^{t|h} \boldsymbol{x}_{k,i}^t \tag{12}$$

$$\boldsymbol{P}_k^{t|h} = \left(1 - \sum_{i=1}^{m_k} \xi_i^{t|h}\right) \boldsymbol{P}_{k|k-1}^t + \sum_{i=1}^{m_k} \xi_i^{t|h} \left[\boldsymbol{P}_{k|k-1}^t - \boldsymbol{K}_k^t \boldsymbol{S}_k^t \left(\boldsymbol{K}_k^t\right)^{\mathrm{T}}\right] \tag{13}$$

where $\boldsymbol{x}_{k,i}^t$ is the updated target state using $\boldsymbol{z}_k^i$ and $\boldsymbol{K}_k^t$ is the filter gain.

- Step 4: The target existence probability, the target state, and the error covariance for track is approximated as [11]

$$r_k^t = \sum_{h=1}^{N_{\mathcal{H}}} r_k^{t|h} \tag{14}$$

$$\boldsymbol{x}_k^t = \frac{1}{r_k^t} \sum_{h=1}^{N_{\mathcal{H}}} r_k^{t|h} \boldsymbol{x}_k^{t|h} \tag{15}$$

$$\boldsymbol{P}_k^t = \frac{1}{r_k^t} \sum_{h=1}^{N_{\mathcal{H}}} r_k^{t|h} \left[\boldsymbol{P}_k^{t|h} + \boldsymbol{x}_k^{t|h} \left(\boldsymbol{x}_k^{t|h}\right)^{\mathrm{T}} - \boldsymbol{x}_k^t \left(\boldsymbol{x}_k^t\right)^{\mathrm{T}}\right] \tag{16}$$

where $N_{\mathcal{H}}$ denotes and the number of all association hypothesis events.

## 4. Evolutionary-Optimization-Based Joint Integrated Probabilistic Data Association

### *4.1. Motivations*

In automatic tracking systems, the target number may vary with time. In this paper, we assume that the targets are not discriminated from each other. Under these assumptions, the target states can be described as RFSs, in which the points are unordered and random [17]. It was shown that the posterior density can be changed into the other density within its RFS family [10]. Therefore, we propose to optimize the ordered posterior density within its RFS family to improve the tracking accuracy.

By reducing the error covariance, the problem of overlapping tracking gates can be effectively alleviated [11]. A scalar measure of the error covariance is the trace for the covariance matrix, as follows [11]

$$\begin{aligned} C_k &= \sum_{t=1}^{n_k} \mathrm{trace}(\boldsymbol{P}_k^t) \\ &= \sum_{t=1}^{n_k} \sum_{h=1}^{N_{\mathcal{H}}} \frac{r_k^{t|h}}{r_k^t} \mathrm{trace}\left(\boldsymbol{P}_k^{t|h} + \boldsymbol{x}_k^{t|h} (\boldsymbol{x}_k^{t|h})^T - \boldsymbol{x}_k^t (\boldsymbol{x}_k^t)^T\right) \\ &= \sum_{t=1}^{n_k} \frac{1}{r_k^t} \sum_{h=1}^{N_{\mathcal{H}}} r_k^{t|h} \left[\mathrm{trace}\left(\boldsymbol{P}_k^{t|h}\right) + \|\boldsymbol{x}_k^{t|h} - \boldsymbol{x}_k^t\|^2\right] \end{aligned} \tag{17}$$

To alleviate the problem of overlapping tracking gates and improve the tracking accuracy, the trace for the covariance matrix Equation (17) is used as the cost function for the optimization of the posterior density.

### 4.2. Evolutionary Optimization of the Posterior Density

For the optimization of the posterior density, the target permutations under different association hypothesis events are reordered to minimize the cost function using Equation (17). We use the evolutionary algorithm [26,27] to implement the optimization. The specific steps of the proposed EOJIPDA method are as follows.

- Step 1: It is necessary to generate the initial population for the evolutionary optimization.

We enumerate all the possible permutations of the targets and assign a unique index to each permutation. For example, when the number of targets is $n_k = 3$, the number of all possible permutations of the targets is $N_P = n_k! = 3! = 6$. The set of the indexes for these permutations is denoted as

$$I = \{I_1, I_1, \ldots, I_{n_k!}\} \tag{18}$$

The indices within the set $I$ are randomly assigned to each association hypothesis event. An assignment indicates a possible solution for the evolutionary algorithm. Such a solution is named as a chromosome, as follows

$$chromosome = [C_1, C_2, \ldots C_{N_{\mathcal{H}}}] \tag{19}$$

where the element $C_j \in I$ and $j = 1, 2, \ldots, N_{\mathcal{H}}$. We randomly perform the assignments and generate $N^{pop}$ chromosomes. These chromosomes are used as the initial population, i.e., the first generation, for the evolutionary optimization.

- Step 2: The value of the cost function forms the fitness value for each solution.

The fitness function is used to test how well the chromosome solves the problem. All the solutions in the population are sorted based on their fitness values.

- Step 3: The offspring solutions are generated using the selection, crossover and mutation operators.

The selection operator chooses some of the chromosomes for reproduction. For the crossover operator, we use the single-point crossover. The genes of the two-parent chromosome are interchanged after a random selected point. Along with the crossover, the mutation is performed. We random select one point and change it into a random selected point from the set using Equation (18).

The new population with all the parents and offspring is sorted again based on their fitness values and the population size is decreased to $N^{pop}$ by eliminating all the lower rank solutions.

- Step 4: Back to Step 3 and the cycle repeats.

For each generation, we record the chromosome with the highest fitness (along with the value of the fitness). The cycle is repeated until the fitness value of the "best-so-far" chromosome stabilizes.

### 4.3. Illustrative Example

To illustrate the procedure of the proposed EOJIPDA method, a one-dimensional example is used in this subsection. We assume that two Gaussian distributed targets generate two detections and there are two association hypothesis events $\theta_1$ and $\theta_2$. The density of target $t$ corresponding to $\theta_h$ is denoted as $p^{t|h}(x) = \left\{r^{t|h}, \mathcal{N}(x^{t|h}, P^{t|h})\right\}$. The initial posterior probability density is given as

$$\begin{aligned} \theta_1 : p^{1|1}(x) = \{0.1,\ N(1.0, 1.0)\},\ p^{2|1}(x) = \{0.4,\ N(5.0, 1.0)\} \\ \theta_2 : p^{1|2}(x) = \{0.5,\ N(4.0, 1.0)\},\ p^{2|2}(x) = \{0.5,\ N(2.0, 1.0)\} \end{aligned} \tag{20}$$

Using Equations (14)–(16), the approximated probability density for each track is described by

$$p^1(x) = \{0.60,\ N(3.50, 2.25)\},\ p^2(x) = \{0.9,\ N(3.33, 3.22)\} \tag{21}$$

The posterior probability density of the joint track state and its approximated probability density of the estimated track state are shown in Figures 1a and 1b, respectively. Obviously, Figure 1b is very different with Figure 1a. Therefore, the approximation is less accurate and the trace of the covariance matrix in Equation (17) of the estimated covariance matrix is $C_0 = 2.25 + 3.22 = 5.47$.

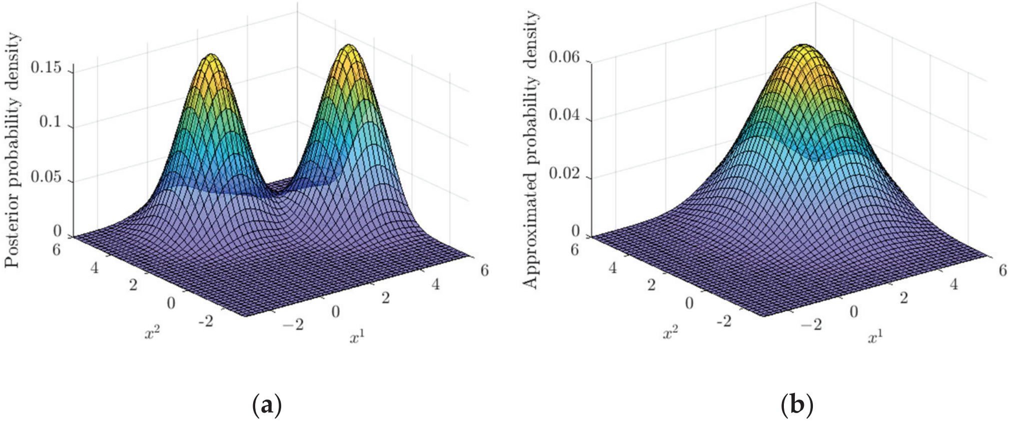


**Figure 1.** The posterior probability density and its approximated probability density of iteration 0: (**a**) posterior probability density of iteration 0 (**b**) approximated probability density of iteration 0.

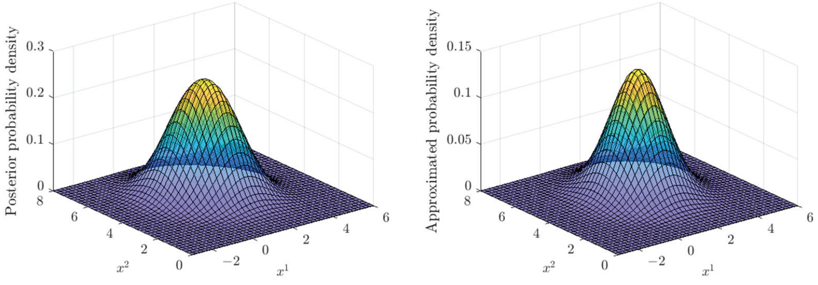
To improve the accuracy of the approximation, we perform the evolutionary optimization of the posterior probability density. Five iterations are performed and the resulting posterior probability density is given as

$$\begin{array}{l} \theta_1 : p^{1|1}(x) = \{0.1,\ N(1.0, 1.0)\},\ p^{2|1}(x) = \{0.4,\ N(5.0, 1.0)\} \\ \theta_2 : p^{1|2}(x) = \{0.5,\ N(2.0, 1.0)\},\ p^{2|2}(x) = \{0.5,\ N(4.0, 1.0)\} \end{array} \tag{22}$$

Using Equations (14)–(16), the approximated probability density for each track is described by

$$p^1(x) = \{0.60,\ N(1.83, 1.14)\},\ p^2(x) = \{0.9,\ N(4.44, 1.25)\} \tag{23}$$

The resulting posterior probability density and its approximated probability density are plotted in Figure 2a,b, respectively. It is obvious that the similarity between Figure 2a,b is much higher than that between Figure 1a,b. The trace of the covariance matrix in Equation (17) of the estimated covariance matrix is reduced to $C_0 = 1.14 + 1.25 = 2.39$. This indicates that the accuracy of the approximation and the state estimation is greatly improved. The results show a good agreement with the theoretical analysis.



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

**Figure 2.** The posterior probability density and its approximated probability density of iteration 5: (**a**) posterior probability density of iteration 5 (**b**) approximated probability density of iteration 5.

## 5. Numerical Simulation and Results

We demonstrate the performance of the proposed EOJIPDA filter using two challenging MTT scenarios in the two-dimensional surveillance area. At time $k$, the state of the moving target is represented by the state vector

$$\boldsymbol{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^{\mathrm{T}} \tag{24}$$

where $[x_k, y_k]^{\mathrm{T}}$ is the target position and $[\dot{x}_k, \dot{y}_k]^{\mathrm{T}}$ is the target velocity. The transition matrix of the motion model in Equation (1) is

$$\boldsymbol{F} = \boldsymbol{I}_2 \otimes \begin{bmatrix} 1 & T_{\mathrm{s}} \\ 0 & 1 \end{bmatrix} \tag{25}$$

where $\otimes$ is the Kroneker product, $\boldsymbol{I}_2$ is a $2 \times 2$ identity matrix, and $T_{\mathrm{s}}$ is the sampling interval. The covariance of the Gaussian process noise $\boldsymbol{u}_k$ is

$$\boldsymbol{Q} = \boldsymbol{I}_2 \otimes q \begin{bmatrix} \frac{T_{\mathrm{s}}^3}{3} & \frac{T_{\mathrm{s}}^2}{2} \\ \frac{T_{\mathrm{s}}^2}{2} & T_{\mathrm{s}} \end{bmatrix} \tag{26}$$

where $q$ is a tuning parameter. In both scenarios, a sensor is used to collect the measurements in the surveillance area. For simplicity, it is assumed that the measurements collected by the sensor represent the positions of the targets. Therefore, the observation matrix of the measurement model in Equation (3) is

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{27}$$

The covariance of the Gaussian measurement noise $\boldsymbol{w}_k$ is

$$\boldsymbol{R} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{28}$$

where $\sigma_x$ and $\sigma_y$ are the standard deviations of the measurement noise in $x$ and $y$ coordinates, respectively.

The sampling interval of the sensor is fixed as $T_{\mathrm{s}} = 1$ s and the detection probability is $P_D = 0.92$. The gating threshold is $G = 9.21$ [28], which corresponds to a two-dimensional gating probability of $P_W = 0.99$. The coefficients of the Markov chain one model in Equation (5) are borrowed from [9], where $p_{11} = 0.98$ and $p_{21} = 0$. The tracks are terminated if the target existence probability falls below the termination threshold and are confirmed if the probability exceeds the confirmation threshold. The confirmation and termination thresholds are $P_C = 0.83$ and $P_T = 0.0909$ [9], respectively. The tracking performance of the filter is measured by the optimal sub-pattern assignment (OSPA) [29] multi-target miss distance, which measures the error between the estimated and true state and is widely used [17–20,30]. If $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, $Y = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_m\}$, and $n \geq m$, the OSPA multi-target miss distance is defined as [29]

$$d_{p,c}^{\text{OSPA}}(X, Y) := \left(\frac{1}{n}\left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c)}(\mathbf{x}_i, \mathbf{y}_{\pi(i)})^p + c^p(n-m)\right)\right)^{1/p} \tag{29}$$

where $d^{(c)}(x, y) := \min(c, \|x - y\|)$, $c > 0$ is the cut-off parameter, $p \geq 1$ is an order parameter and $\pi$ is a permutation function in the set of permutations $\Pi_n$. If $n < m$, $d_{p,c}^{\text{OSPA}}(X, Y) := d_{p,c}^{\text{OSPA}}(Y, X)$. All the experiments are tested in MATLAB R2010a and implemented on a computer with a 3.40 GHz processor.

### *5.1. Scenario 1*

We consider the tracking of two targets in this scenario, whose trajectories are shown in Figure 3. The track coalescence of the JPDA and JIPDA filters can be clearly observed using this scenario; hence, this scenario has been widely used [10–15]. In Figure 3, the parameters are selected as: $\phi = \pi/3$, $d = 0.5$ m, and $l_1 = l_2 = 10$ m. It is assumed that the speed of the target stays constant as $v = 1$ m/s. The parameters for the clutter measurements are $\lambda = 0.36 \times 10^{-3}\ \text{m}^{-2}$ and $|\text{FOV}| = 1.4 \times 10^3\ \text{m}^2$, giving an average of five clutter returns per scan. The standard deviations of the measurement noise are $\sigma_x = \sigma_y = 0.1$ m .

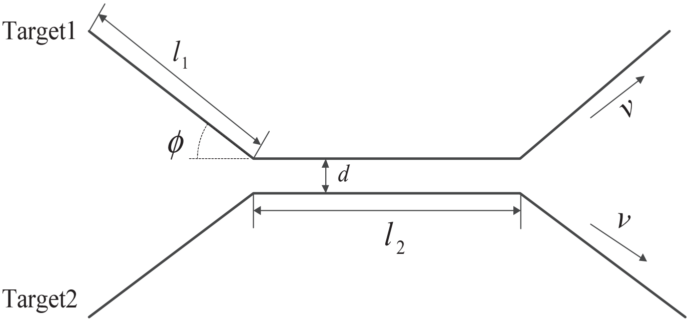


**Figure 3.** Simulated scenario 1.

The estimated target positions for a single run of the JIPDA and EOJIPDA filters are shown in Figures 4a and 4b, respectively. It can be observed that the serious track coalescence problem happens in Figure 4a. When the targets are closely spaced, their position estimates coalesce. Although the targets move away from each other later, the JIPDA fails to detect their separation. Compared with the JIPDA filter, the EOJIPDA filter can accurately estimate the target positions, as shown in Figure 4b.

The average OSPA distances over 100 Monte Carlo trails for the JIPDA, exact nearest neighbor JIPDA (ENNJIPDA), JIPDA*, and EOJIPDA filters are shown in Figure 5. Due to the track coalescence problem, the OSPA distance of the JIPDA filter is worse than in the comparative algorithms. Although the ENNJPDA filter was proposed to overcome the coalescence problem, it is sensitive to clutter and missed detections. Compared with the JIPDA* filter, the EOJIPDA filter uses more information about the posterior density and performs better in terms of the OSPA error.

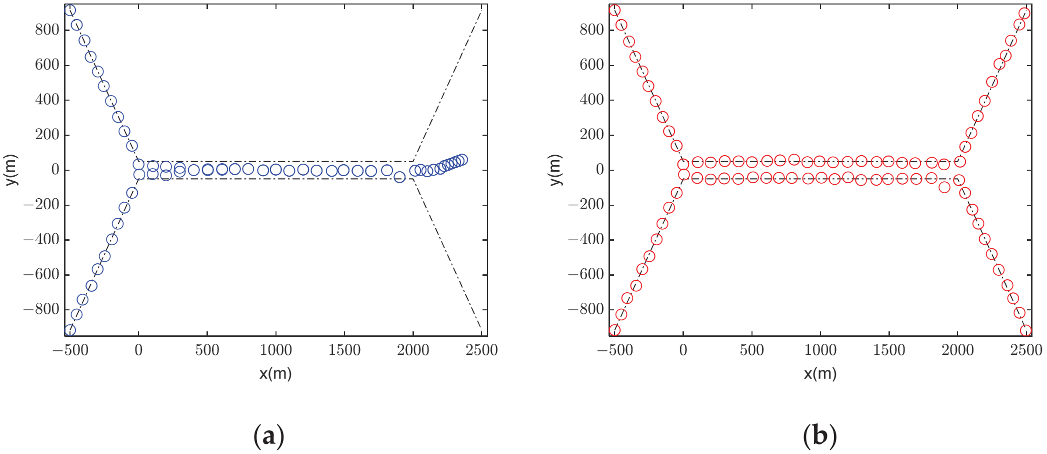


**Figure 4.** Estimated target positions for Scenario 1, in which circles represent estimated positions and dotted lines represent true trajectories: (**a**) Output of the JIPDA filter; (**b**) output of the EOJIPDA filter.

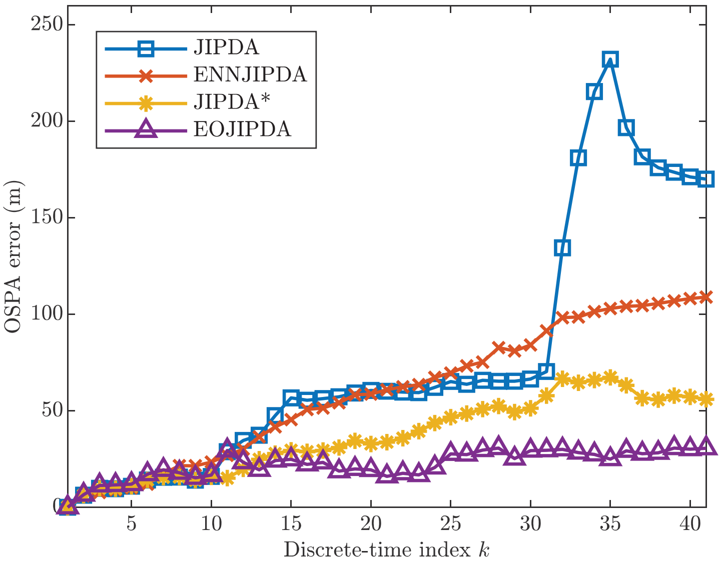


**Figure 5.** Average OSPA distances for Scenario 1. JIPDA* denotes a coalescence-avoiding version of JIPDA, as shown in [16].

*5.2. Scenario 2*

The tracking of three targets is studied in this scenario. The trajectories of the targets cross each other at a crossing point, as shown in Figure 6. Each target keeps a constant speed, $v = 1$ m/s. The parameters used in the simulations are $\phi_1 = \phi_2 = \pi/8$ and $l_1 = l_2 = 100$ m, $\sigma_x = \sigma_y = 10$ m, $\lambda = 2.5 \times 10^{-3}$ m$^{-2}$ , and $|\text{FOV}| = 2 \times 10^3$ m$^2$. The two-point difference [24] is used to initiate tracks automatically and the maximum possible speed is assumed to be 200 m/s.

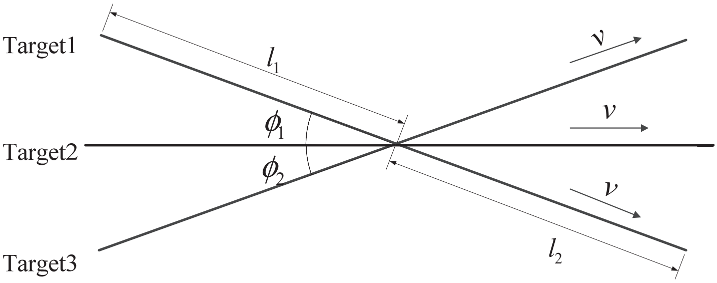


**Figure 6.** Simulated scenario 2.

The estimated target positions for a single run of the JIPDA and EOJIPDA filters are shown in Figures 7a and 7b, respectively. It can be observed from Figure 7 that it takes some time for the JIPDA and EOJIPDA filters to localize targets. When the targets are spaced close to each other, the JIPDA filter cannot accurately estimate the target positions and the tracks tend to coalesce. Compared with the JIPDA filter, the EOJIPDA filter can effectively overcome the track coalescence problem and provide position estimates that are close to the true values.

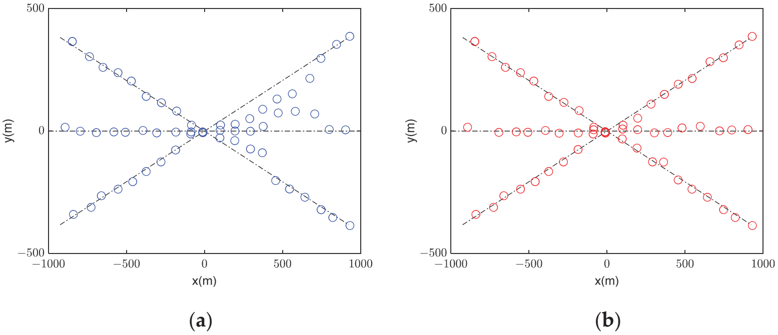


**Figure 7.** Estimated target positions for Scenario 2, in which circles represent estimated positions and dotted lines represent true trajectories: (**a**) Output of the JIPDA filter; (**b**) output of the EOJIPDA filter.

The average OSPA distances for the different tracking methods are shown in Figure 8. The tracking errors of all the methods are high at the initial time steps because the initial states of the targets are assumed to be unknown in this scenario. Since the track coalescence problem happens in the JIPDA filter, it can be observed that the tracking error of the JIPDA filter is high after the crossing point (at time $k = 11$). The ENNJIPDA and JIPDA* filters can effectively reduce the tracking error of the JIPDA filter. Nevertheless, the error performance of the proposed EOJIPDA filter is better than that of other filters for almost the entire scenario.

Table 2 shows the accumulated number of confirmed tracks at the end point of the tracking by 100 MC trails for the JIPDA filter, the ENNJIPDA filter, the JIPDA* filter and the EOJIPDA filter. The results for the different clutter rates are provided. It can be seen that the challenging tracking environment makes all the filters lose some tracks. Nevertheless, the performance of the EOJIPDA filter is superior to others under different clutter rates. In other words, the better tracking performance of the EOJIPDA filter makes the detection of target existence more reliable.

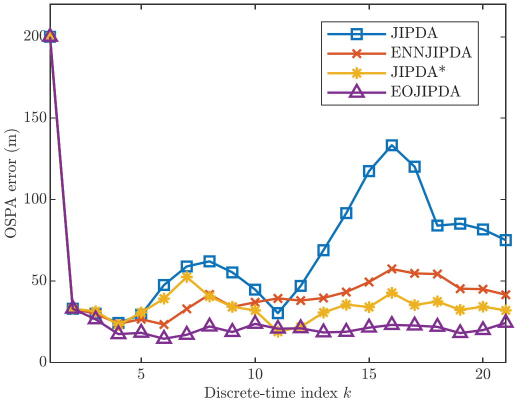


**Figure 8.** Average OSPA distances for Scenario 2.

**Table 2.** Number of confirmed targets.

| Clutter Rate | JIPDA | ENNJIPDA | JIPDA* | EOJIPDA |
|---|---|---|---|---|
| $r = 3$ | 241 | 287 | 293 | 299 |
| $r = 6$ | 238 | 281 | 290 | 297 |
| $r = 9$ | 230 | 279 | 283 | 294 |

## 6. Conclusions

The JIPDA filter an effective method for MTT, but it suffers from the serious track coalescence problem. To improve the tracking performance of the JIPDA filter, a novel MTT filter, named EOJIPDA, was proposed in this paper. We first attempted to use the evolutionary computation technique to optimize the posterior density of the JIPDA filter. When the target identity was irrelevant, we modeled the posterior density optimization problem as an evolutionary computation problem and the trace of the covariance matrix is used as the cost function. Through an indicative example, it was shown that the evolutionary computation can effectively reduce the value of the cost function, improving the accuracy of the Gaussian approximation and the state estimation.

The simulation results show that the EOJIPDA filter effectively avoids track coalescence and performs better than the traditional algorithms in terms of the OSPA error. Future work will investigate the application of more computational intelligence strategies, such as particle swarm optimization and the Cuckoo search algorithm, to improving the performances of traditional tracking methods.

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

**Funding:** This research received no external funding.

**Data Availability Statement:** In this work, we have used the free RFS MATLAB code at http://ba-tuong.vo-au.com/codes.html.

**Acknowledgments:** This work was supported in part by the National Natural Science Foundation of China, under grant numbers 62007022 and 61906146; the Natural Science Foundation of Shaanxi Province, under grant number 2021JQ-209; and the Fundamental Research Funds for the Central Universities, under grant numbers GK202103082 and JB210210.

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

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| MTT | multi-target tracking |
| MHT | multiple hypothesis tracking |
| JPDA | joint probabilistic data association |
| ENNJPDA | exact nearest-neighbor JPDA |
| ENNJIPDA | exact nearest-neighbor JIPDA |
| SJPDA | set JPDA |
| NNSJPDA | nearest-neighbor SJPDA |
| JIPDA | joint integrated probabilistic data association |
| FISST | finite set statistics |
| PHD | probability hypothesis density |
| CPHD | cardinalized PHD |
| MB | multi-Bernoulli |
| LMB | label multi-Bernoulli |
| GLMB | generalized label multi-Bernoulli |
| RFS | random finite set |
| GM | Gaussian mixture |
| SMC | sequential Monte Carlo |
| EOJIPDA | joint integrated probabilistic data association |
| OSPA | optimal sub-pattern assignment |
| List of mathematical symbols | |
| $\boldsymbol{x}_k$ | state vector at time $k$ |
| $\boldsymbol{F}_k$ | transition matrix at time $k$ |
| $\mathcal{N}(\boldsymbol{m}, \boldsymbol{P})$ | Gaussian probability density with mean $\boldsymbol{m}$ and covariance $\boldsymbol{P}$ |
| $\boldsymbol{z}_k$ | measurement vector at time $k$ |
| $\boldsymbol{H}_k$ | observation matrix at time $k$ |
| $P_D$ | detection probability |
| $\boldsymbol{P}_k$ | covariance matrix at time $k$ |
| $r_k$ | target existence probability at time $k$ |
| $p_{11}$,$p_{21}$ | Markov chain coefficients |
| $\boldsymbol{z}_{k\|k-1}$ | predicted measurement vector from time $k$-1 to time $k$ |
| $\boldsymbol{S}_k$ | innovation covariance |
| $d_z$ | measurement dimension |
| $G$ | gating threshold |
| $\theta_h$ | association hypothesis event |
| $P_W$ | gating probability |
| $V_k$ | cluster volume |
| $\boldsymbol{K}_k$ | filter gain |
| $N_{\mathcal{H}}$ | number of all association hypothesis events |
| trace$(\boldsymbol{P})$ | trace for the covariance matrix $\boldsymbol{P}$ |

## References

1. Anjos, J.C.S.D.; Gross, J.L.G.; Matteussi, K.J.; González, G.V.; Geyer, C.F.R. An algorithm to minimize energy consumption and elapsed time for IoT workloads in a hybrid architecture. *Sensors* **2021**, *21*, 2914. [CrossRef] [PubMed]
2. Martins, J.A.; Ochôa, I.S.; Silva, L.A.; Mendes, A.S.; González, G.V.; Santana, J.D.; Leithardt, V.R.Q. PRIPRO: A comparison of classification algorithms for managing receiving notifications in smart environments. *Appl. Sci.* **2020**, *10*, 502. [CrossRef]
3. Baser, E.; McDonald, M.; Kirubarajan, T.; Efe, M. A joint multitarget estimator for the joint target detection and tracking filter. *IEEE Trans. Signal Process.* **2015**, *63*, 3857–3871. [CrossRef]
4. Mallick, M.; Vo, B.N.; Kirubarajan, T.; Arulampalam, S. Introduction to the issue on multitarget tracking. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 373–375. [CrossRef]
5. Severson, T.A.; Paley, D.A. Distributed multitarget search and track assignment with consensus based coordination. *IEEE Sens. J.* **2015**, *15*, 864–875. [CrossRef]
6. Reid, D. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Cont.* **1979**, *24*, 843–854. [CrossRef]
7. Kurien, T. Issues in the design of practical multitarget tracking algorithms. In *Multitarget-Multisensor Tracking: Advanced Applications*; Bar-Shalom, Y., Ed.; Artech House: Norwood, MA, USA, 1990; pp. 43–83.

8. Fortmann, T.E.; Bar-Shalom, Y.; Scheffe, M. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J. Ocean. Eng.* **1983**, *8*, 173–184. [CrossRef]
9. Mušicki, D.; Evans, R. Joint integrated probabilistic data association: JIPDA. *IEEE Trans. Aerosp. Electron. Syst.* **2004**, *40*, 1093–1099. [CrossRef]
10. Svensson, L.; Svensson, D.; Guerriero, M.; Willett, P. Set JPDA filter for multitarget tracking. *IEEE Trans. Signal Process.* **2011**, *59*, 4677–4691. [CrossRef]
11. Zhu, Y.; Wang, J.; Liang, S.; Wang, J. Covariance control joint integrated probabilistic data association filter for multi-target tracking. *IET Radar Sonar Nav.* **2019**, *13*, 584–592. [CrossRef]
12. Fitzgerald, R.J. Development of practical PDA logic for multitarget tracking by microprocessor. In *Multitarget-Multisensor Tracking: Advanced Applications*; Bar-Shalom, Y., Ed.; Artech House: Norwood, MA, USA, 1990; pp. 1–23.
13. Blom, H.; Bloem, E. Probabilistic data association avoiding track coalescence. *IEEE Trans. Autom. Cont.* **2000**, *45*, 247–259. [CrossRef]
14. Zhu, Y.; Wang, J.; Liang, S. Efficient joint probabilistic data association filter based on Kullback-Leibler divergence for multi-target tracking. *IET Radar Sonar Nav.* **2017**, *11*, 1540–1548. [CrossRef]
15. Zhu, Y.; Liang, S.; Wu, X.J.; Yang, H.H. A random finite set based joint probabilistic data association filter with non-homogeneous Markov chain. *Front. Inf. Technol. Electron. Eng.* **2021**, *22*, 1114–1126. [CrossRef]
16. Blom, H.; Bloem, E.; Mušicki, D. JIPDA*: Automatic target tracking avoiding track coalescence. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 962–974. [CrossRef]
17. Mahler, R. *Statistical Multisource Multitarget Information Fusion*; Artech House: Norwood, MA, USA, 2007.
18. Mahler, R. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 1152–1178. [CrossRef]
19. Mahler, R. PHD filters of higher order in target number. *IEEE Trans. Aerosp. Electron. Syst.* **2007**, *43*, 1523–1543. [CrossRef]
20. Vo, B.T.; Vo, B.N.; Cantoni, A. The cardinality balanced multi-target multi-Bernoulli filter and its implementations. *IEEE Trans. Signal Process.* **2009**, *57*, 409–423. [CrossRef]
21. Reuter, S.; Vo, B.T.; Vo, B.N.; Dietmayer, K. The labeled multi-Bernoulli filter. *IEEE Trans. Signal Process.* **2014**, *62*, 3246–3260. [CrossRef]
22. Vo, B.N.; Vo, B.T.; Phung, D. Labeled random finite sets and the Bayes multi-target tracking filter. *IEEE Trans. Signal Process.* **2014**, *62*, 6554–6567. [CrossRef]
23. Vo, B.T.; Vo, B.N. Labeled random finite sets and multi-target conjugate priors. *IEEE Trans. Signal Process.* **2013**, *61*, 3460–3475. [CrossRef]
24. Blackman, S.; Popoli, R. *Design and Analysis of Modern Tracking Systems*; Artech House: Norwood, MA, USA, 1999.
25. Breidt, F.J.; Carriquiry, A.L. Highest density gates for target tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 47–55. [CrossRef]
26. Abraham, A.; Nedjah, N.; Mourelle, L.M. Evolutionary computation: From genetic algorithms to genetic programming. In *Genetic Systems Programming: Theory and Experiences*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 13, pp. 1–20. [CrossRef]
27. Gong, M.G.; Li, H.; Meng, D.Y.; Miao, Q.G. Decomposition-Based Evolutionary Multi-objective Optimization to Self-paced Learning. *IEEE Trans. Evolut. Comput.* **2019**, *23*, 288–302. [CrossRef]
28. Bar-Shalom, Y.; Chang, K.C.; Blom, H.A.P. *Automatic Track Formation in Clutter with a Recursive Algorithm. Multitarget-Multisensor Tracking*; Artech House: Norwood, MA, USA, 1990; pp. 25–42.
29. Schuhmacher, D.; Vo, B.T.; Vo, B.N. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. Signal Process.* **2008**, *56*, 3447–3457. [CrossRef]
30. Zhu, Y.; Wang, J.; Liang, S. Multi-Objective Optimization Based Multi-Bernoulli Sensor Selection for Multi-Target Tracking. *Sensors* **2019**, *19*, 980. [CrossRef] [PubMed]

*electronics*

MDPI

*Article*

# A Novel Progressive Image Classification Method Based on Hierarchical Convolutional Neural Networks

**Cheng Li [1], Fei Miao [1,*] and Gang Gao [2]**

[1] Department of Ultrasound, Ruijin Hospital, Shanghai Jiao Tong University School of Medicine, Shanghai 200025, China; lc12685@rjh.com.cn
[2] Shanghai Yiran Health Consulting Co., Ltd., Shanghai 201821, China; xiefei@nwpu.edu.cn
* Correspondence: mf11066@rjh.com.cn

**Abstract:** Deep Neural Networks (DNNs) are commonly used methods in computational intelligence. Most prevalent DNN-based image classification methods are dedicated to promoting the performance by designing complicated network architectures and requiring large amounts of model parameters. These large-scale DNN-based models are performed on all images consistently. However, since there are meaningful differences between images, it is difficult to accurately classify all images by a consistent network architecture. For example, a deeper network is fit for the images that are difficult to be distinguished, but may lead to model overfitting for simple images. Therefore, we should selectively use different models to deal with different images, which is similar to the human cognition mechanism, in which different levels of neurons are activated according to the difficulty of object recognition. To this end, we propose a Hierarchical Convolutional Neural Network (HCNN) for image classification in this paper. HCNNs comprise multiple sub-networks, which can be viewed as different levels of neurons in humans, and these sub-networks are used to classify the images progressively. Specifically, we first initialize the weight of each image and each image category, and these images and initial weights are used for training the first sub-network. Then, according to the predicted results of the first sub-network, the weights of misclassified images are increased, while the weights of correctly classified images are decreased. Furthermore, the images with the updated weights are used for training the next sub-networks. Similar operations are performed on all sub-networks. In the test stage, each image passes through the sub-networks in turn. If the prediction confidences in a sub-network are higher than a given threshold, then the results are output directly. Otherwise, deeper visual features need to be learned successively by the subsequent sub-networks until a reliable image classification result is obtained or the last sub-network is reached. Experimental results show that HCNNs can obtain better results than classical CNNs and the existing models based on ensemble learning. HCNNs have 2.68% higher accuracy than Residual Network 50 (Resnet50) on the ultrasonic image dataset, 1.19% than Resnet50 on the chimpanzee facial image dataset, and 10.86% than Adaboost-CNN on the CIFAR-10 dataset. Furthermore, the HCNN is extensible, since the types of sub-networks and their combinations can be dynamically adjusted.

**Keywords:** computational intelligence; image classification; HCNNs; progressive deep learning; disease screening

**Citation:** Li, C.; Miao, F.; Gao, G. A Novel Progressive Image Classification Method Based on Hierarchical Convolutional Neural Networks. *Electronics* **2021**, *10*, 3183. https://doi.org/10.3390/electronics10243183

Academic Editor: Stefanos Kollias

Received: 1 November 2021
Accepted: 16 December 2021
Published: 20 December 2021

## 1. Introduction

With the development of computer vision technologies, many visual tasks, such as object detection, semantic segmentation, and image classification, have been widely applied in many fields [1–3]. Image classification is one of the most common and important visual tasks [4–6], and a large number of models have been proposed based on traditional machine learning methods and deep learning methods [7–9]. Recently, Convolutional Neural Network(CNN)-based image classification methods, such as AlexNet [10], Visual Geometry Group 16 (VGG16) [11], ResNet [12], and Densely Connected Networks (DenseNet) [13,14], were widely applied in many visual tasks. Generally speaking, the networks with fewer layers usually extract the low-level visual features, while the networks with more layers can extract the more abstract visual features.

These primary research works focus on how to extract distinguishable local features to improve the image classification performance. In practice, however, there are large numbers of various types of objects, and many images suffer from poor illumination conditions, varying degrees of occlusion, similarities between objects, and so on. It is difficult to accurately classify all images by a consistent model, which presents great challenges to image classification [15,16]. For human beings, different types of objects are recognized through different processes, and people tend to quickly make judgments on easy-to-recognize objects based on their own subjective and objective cognition or prior knowledge. Meanwhile, people need further analysis and understanding for relatively difficult-to-recognize objects, and may further perform information abstraction and knowledge reasoning. Therefore, we contend that there are meaningful differences between images, and various models encounter various difficulties when attempting to accurately classify them. For example, images with appropriate lighting conditions are more easily classified correctly by the model than those with strong or weak lighting conditions; it is easier to perform disease screening on medical images for prominent lesions [17,18]. Therefore, we should select the appropriate networks according to the particular tasks. However, in most traditional CNN-based methods, all images need to be sent to the same classification process, which neglects the differences in discrepant classification difficulties for different images [19–21].

Inspired by the mechanism of human cognition and the fact that different images present different levels of cognitive difficulty, we design a hierarchical integrated deep learning model named HCNN. The HCNN treats multiple CNNs as sub-networks and uses them progressively for feature extraction [22,23]. Specifically, the simple sub-networks are used to extract visual features for the images that are easy to classify accurately. Moreover, the complex sub-networks are used to extract the more abstract visual features, which are more suitable for the images which are more difficult to accurately classify. The final classification results are obtained by integrating the results of these sub-networks. Most existing models integrate multiple CNNs by fusing the high-level feature/decision of the CNNs to obtain a final result. Our HCNN selectively extracts the composite features of multiple sub-networks in different levels, which is more reasonable and complies with the process of human cognition.

Furthermore, the multi-class joint loss is designed to offer the features of the samples within the same category higher similarity, while the similarity between the features of different categories is made as low as possible. Gradient descent is used to train the entire network end-to-end. Finally, several experiments are conducted on a medical image dataset, two common image classification datasets (CIFAR-10, CIFAR-100 [24]), and a chimpanzee dataset [25]. The comparison experimental results show that the HCNN achieves superior performance to the existing related models. Moreover, ablation experiments prove that our model's performance is superior to that of each single network and combinations of several sub-networks. In addition, it is worth noting that the HCNN has good scalability, since the types and combinations of CNN modules can be dynamically adjusted depending on the specific tasks involved.

The main contributions of this paper are as follows:

(1) We propose a progressive image classification model, named HCNN, which can progressively use its sub-network modules (with different depths of network layers) to extract different levels of visual features from images, while the classification results of different images are output by corresponding sub-network modules. In brief, the HCNN can use the sub-network modules with fewer network layers to quickly yield image classification results for the images that are easy to classify accurately, while the images that are difficult to classify accurately need to pass through more complex sub-network modules.

(2) A multi-class joint loss is designed to reduce the distance between the features of samples within the same category, while increasing the distance between the features of

samples in different categories. In addition, gradient descent is used for the entire model training end-to-end.

(3) The performance and scalability of the HCNN are verified on four image classification datasets. The comparison and ablation experimental results show that the HCNN achieves significant performance improvements compared with existing models and combinations of several sub-networks.

This paper is organized as follows. In Section 2, we review the related image classification models, ensemble learning models, and metric learning models and describes their relationships with our model. In Section 3, we elaborate the basic framework and loss functions of HCNNs and give the model implementation process in the test stage. In Section 4, we compare HCNNs and eight related methods on our own ultrasonic image dataset and three public image datasets. We also perform validation experiments to further analyze the HCNNs. The final conclusion is given in Section 5.

## 2. Related Work

Image classification is one of the most important visual tasks in computer vision. Due to the rapid development of deep learning technologies and its superior performance in computer vision, image classification methods based on DNNs have become increasingly mature. To accurately classify images, various types of artificial visual features are designed, and the visual features are automatically learned by DNNs. Related classifiers are then used to distinguish the categories of the images. To date, a large number of deep learning-based image classification methods have been proposed [26,27] and have been widely used in different computer vision tasks. In addition, several improved models have been successively proposed to improve the image classification performance. Xi et al. [28] proposed a parallel neural network by combining texture features. This model can extract features that are highly correlated with facial changes, and thus achieves better performance in facial expression recognition. Hossain et al. [29] developed an automatic date fruit classification system to satisfy the interest of date fruit consumers. Goren et al. [30] collected the street images taken by roadside cameras to form a dataset, and then designed a CNN to check the vacancy in the collected dataset. To form an efficient classification mechanism that integrates feature extraction, feature selection, and a classification model, Yao et al. [31] proposed an end-to-end image classification method based on an aided capsule network and applied it to traffic image classification. An image classification framework for securing against indistinguishable plaintext attacks was proposed by Hassan et al. [32]. This framework performs a secure image classification on the cloud without the need for constant device interaction. To solve the multi-class classification problems, Vasan et al. [33] proposed a new method to convert raw malware binaries into color images, which are used by the fine-tuned CNN architecture to detect and identify malware families.

A single CNN may be impacted by gradient disappearance, gradient explosion, and other similar factors, while network models based on ensemble learning have better immunity to these adverse factors due to the cooperative complementation of multiple CNNs. For example, Ciregan et al. [34] designed a method by utilizing multiple CNNs, which are trained by using the same training datasets. These trained CNNs are then used to obtain multiple prediction results, which are in turn fused to obtain the final result. This method employs the simple addition of the predicted results of different CNNs, which it treats in isolation. Frazao et al. [35] assigned different weights to multiple CNNs; the CNNs with better performance have higher weights, and therefore have greater impacts on the final results. An integration of CNNs is used to detect polyps by Tajbakhsh et al. [36]; this approach can accurately identify the specific types of polyps by using their color, texture, and shape features. Ijjina et al. [37] proposed a human action prediction method, which combines several CNNs and uses the best predicted result as the final result. Although these methods use multiple neural network modules to carry out related classification tasks, the modules are independent of each other and the interactions between models are ignored. To solve these problems, Adaboost CNN models have been proposed. For example, Taherkhani et al. [38]

combined several CNN sub-networks based on the Adaboost algorithm. These CNN sub-networks have the same network structure; thus, the transfer learning method can be used between adjacent layers, and the last CNN sub-network module outputs the final results. The model in [38] is unable to selectively and progressively use the CNN sub-networks for feature extraction, and the testing images need to go through all CNN sub-networks to obtain the final results.

A key problem with the semantic understanding of images is that of learning a good metric to measure the similarity between images. Deep metric learning-based methods have been proposed to learn the appropriate similarity measures between pairs of samples, while samples with higher similarities are classified into a single category according to the distances between samples. These approaches have been widely used for image retrieval [39], face recognition [40], and person re-identification [41]. For example, Schroff et al. [42] proposed a face recognition system named FaceNet, and a triplet loss was designed to measure the similarities between samples. Wang et al. [43] proposed a general weighting framework for a series of existing pair-based loss functions by fully considering three similarities for pair weighting, and then collecting and weighting the informative pairs. These metric learning methods focus on optimizing the similarity of image pairs. Furthermore, center loss is proposed by Wen et al. [44] to define a category center for each category, as well as to minimize the distance within one category. Wang et al. [45] proposed an angular loss, which considers the angle relationship to learn a better similarity metric, while the angular loss aims at constraining the angle at the negative point of triplet triangles.

The related works mentioned above mainly involve DNNs, ensemble learning, and metric learning. Meanwhile, there are intrinsic correlations between these fields. In general, ensemble learning needs to use multiple DNN models, and the design of both ensemble learning and DNNs should be on the basis of the theory of metric learning. Specifically, the proposed HCNN is an ensemble learning model based on DNNs for the image classification task, and the multi-class joint loss is designed for the HCNN according to the basic theory of metric learning.

## 3. The Proposed Hierarchical CNNs (HCNNs)

In order to classify different images in real life, we design a hierarchical progressive DNN framework, named Hierarchical CNNs (HCNNs), which consists of several sub-networks. The images need to go through one or more sub-networks so as to obtain a more reliable classification result. In this paper, we refer to the definitions of samples in self-paced learning methods [46]: the samples that are easy for models to identify are defined as easy samples, while the difficult-to-identify samples are denoted as hard samples. In this section, we will describe the overall structure of the HCNN and its loss function. Multiple CNNs are combined to form HCNNs, which can progressively carry out the sub-networks to classify the images; the cross-entropy loss and triple loss are combined for model training to more accurately extract the distinguishing features of the images.

### *3.1. The Model Framework of HCNNs*

Based on the basic concept of ensemble learning, we try to aggregate multiple CNNs into a strong image classification model [1,47,48]. However, unlike traditional ensemble learning methods or Adaboost CNNs [38], which consist of the same type of sub-networks that are indiscriminately trained and tested, our HCNN consists of several different types of CNNs as the sub-networks, and these sub-networks are trained progressively in order. In this paper, we choose Alexnet [10], VGG16 [11], Inception V3 [49], Mobilenet V2 [50], and Resnet-50 [12] as the basic sub-networks (see Figure 1). In addition, there are no limits on the number of sub-networks and their types. At the training stage, images are assigned weights to express the difficulties encountered by models in accurately classifying them. If an image can not be accurately classified by a sub-network, its weight will be increased. Images with updated weights are then input into the next sub-network for extracting more abstract and effective visual features. In this section, we will elaborate on HCNNs in more detail.

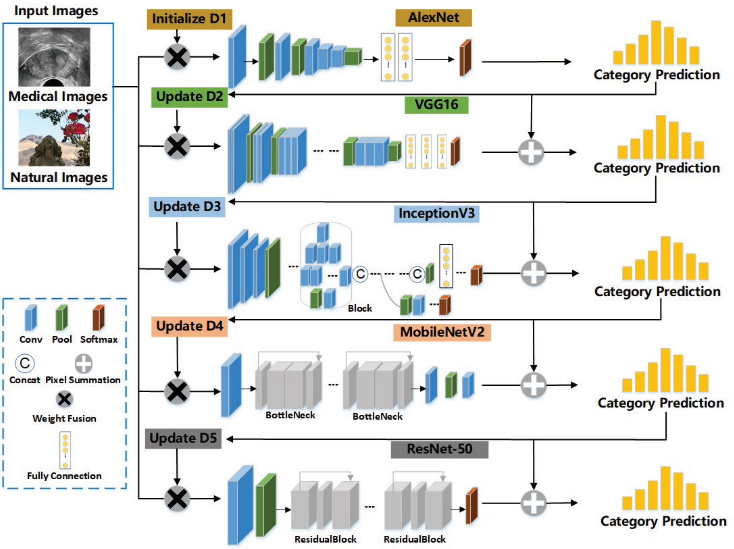


**Figure 1.** An overview of HCNNs. In this paper, the HCNN consists of five sub-networks, i.e., Alexnet, VGG16, Inception V3, Mobilenet, and Resnet-50. Each image sample has its weight for the specific sub-networks. $D_1, \cdots, D_5$ represent the image weights for the sub-networks, respectively. Each sub-network combines the results of the previous sub-networks to make decisions.

Assume that HCNNs have $M$ sub-networks, and they are trained one by one. Let $w_i^m$ be the weight of the $i$-th image for the $m$-th sub-network, and $D^m = \{w_1^m, w_2^m, \ldots, w_n^m\}$. Here, $i \in \{1, 2, \ldots, n\}$, $m \in \{1, 2, \ldots, M\}$, while $n$ is the number of all images in the training dataset.

First of all, the weights of all images need to be initialized. We therefore input all these training images with their initial weights into the first sub-network (Alexnet, $m = 1$) for model training.

$$D^1 = \{w_1^1, w_2^1, \ldots, w_n^1\}, \tag{1}$$

where $w_i^1 = 1/n, i = 1, \cdots, n$. The first sub-network is then trained through multiple iterations. The gradient descent is used to update its parameters in each iteration. Finally, the trained sub-network can give the predictions:

$$y_i^m = G^m(x_i), \tag{2}$$

where $G^m(\cdot)$ represents the $m$-th sub-network, and $y_i^m$ is the predicted label of the $i$-th sample by the $m$-th sub-network $G^m$. Next, we select the samples that meet the condition of $y_i^m \neq t_i$, where $t_i$ is the ground truth of the category label of the $i$-th sample. We further use the following equation to calculate the weighted error rate $\varepsilon^m$ of the $m$-th sub-network $G^m(\cdot)$ on all selected samples in the training set:

$$\varepsilon^m = \sum_{i\_s=1}^{N_{i\_s}} w_{i\_s}^m, \tag{3}$$

where $w^m_{i\_s}$ is the weight of the $i\_s$-th selected samples for $G^m(\cdot)$, and $N_{i\_s}$ is the number of selected samples. Subsequently, $\varepsilon^m$ is used to obtain the weight coefficient $\alpha^m$ of $G^m$, which denotes the importance coefficient of $G^m$ in HCNNs:

$$\alpha^m = \frac{1}{2} log \frac{1-\varepsilon^m}{\varepsilon^m}. \tag{4}$$

As Equation (4) shows, $\alpha^m$ is inversely proportional to $\varepsilon^m$, i.e., with a smaller error rate $\varepsilon^m$, the corresponding sub-network will have larger values of the importance coefficient throughout the whole model. Furthermore, $\alpha^m$ is used to update the weights of the samples to train the next sub-network.

For the images that meet the condition $y^m_i = t_i$, we have

$$w^{(m+1)}_i = w^m_i exp(-\alpha^m). \tag{5}$$

Otherwise,

$$w^{(m+1)}_i = w^m_i exp(\alpha^m). \tag{6}$$

Then,

$$D^{(m+1)} = \{w^{(m+1)}_1, w^{(m+1)}_2, ..., w^{(m+1)}_n\}. \tag{7}$$

Therefore, if the predicted results $y^m_i$ exhibit a high degree of agreement with the true labels $t_i$ of the images, then the weights of the images for the next sub-network decrease; otherwise, their weights increase. We then use the image samples with their updated weights to train the next sub-network for multiple iterations.

For a dataset containing a small number of samples, the initial and updated weights of the samples are applicable to training of HCNNs. However, if the dataset consists of a large number of samples, there is a risk of gradient explosion occurring during model training due to the loss values being too small (possibly even approaching zero); this means that the network parameters cannot be updated normally. To solve this problem, we use the weights of samples to obtain the category weights using Equation (8):

$$C^{(m+1)}_j = \sum_{k\_j=1}^{K_j} w^{(m+1)}_{k\_j}, \tag{8}$$

where $C^{(m+1)}_j$ represents the weight of the $j$-th category for the $(m+1)$-th sub-network, and $w^{(m+1)}_{k\_j}$ is the weight of the $k\_j$-th sample belonging to the $j$-th category, which has $K_j$ samples. We then use $C^{(m+1)}_j$ as the weights of the samples belonging to the $j$-th category (Equation (9)).

$$w'^{(m+1)}_{k\_j} = C^{(m+1)}_j. \tag{9}$$

Therefore, before training each sub-network, we need to update the weights of all samples according to the weights of their corresponding categories. The sub-network will then pay more attention to the samples with larger weights.

HCNN is a scalable model, and its architecture is illustrated in Figure 1. In addition, HCNN enhances the correlation between different sub-networks by transmitting the feature vectors and the sample weights in the previous sub-network to the next sub-network.

### 3.2. Multi-Class Joint Loss in HCNNs

During model training, we constantly updated the weights of the image categories and the images to express the difficulties encountered by the model. We then needed to design the loss function, which can guide the model to extract the specific visual features from different images. In addition, this loss function should attempt to make the difference in the visual features within the same category as small as possible, while the difference in the visual features in different categories should be as large as possible.

Cross-entropy loss with category weights. The cross-entropy function $L_C$ is a classic and commonly used loss function. In this paper, to enable the HCNN to select its corresponding sub-networks and therefore extract the visual features in different levels, a category weight is assigned to each image category; subsequently, the new cross-entropy loss with category weights can be expressed by the following equation:

$$L_C^{'(m+1)} = C_j^m L_C^{(m+1)}. \tag{10}$$

Here, $L_C^{'(m+1)}$ is the cross-entropy loss with category weights for the $(m+1)$-th sub-network, and $L_C^{(m+1)}$ is the traditional cross-entropy loss.

Weighted triplet loss. For image classification, the problem may arise that there may be less similarity between images within the same category, while there is more similarity between images in different categories; as a result, it is difficult to effectively improve the image classification performance. The triplet loss can guide models to learn the visual features to further cluster the samples within the same category and separate the samples of different categories. Therefore, we use a weighted triplet loss in each sub-network. This guides HCNNs to extract more discriminative visual features between the samples of different categories, as shown in Figure 2.

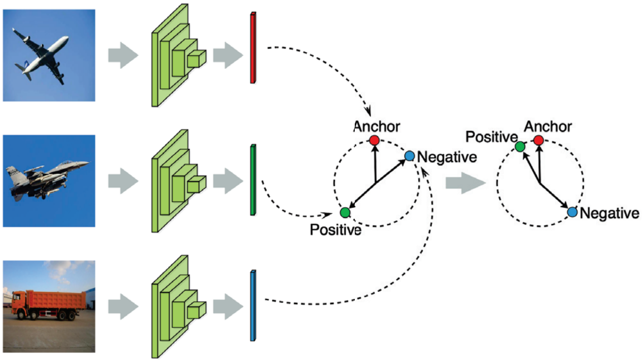


**Figure 2.** An illustration of the influence of triplet loss on visual feature learning.

Assume that we have a series of image samples $\{x_1, x_2, ..., x_n\}$, and $\{y_1, y_2, ..., y_n\}$ are their true labels. We then define an anchor image $u^a$, a positive image sample $u^+$, and a negative image sample $u^-$. More specifically, $u^a$ is an image in one category, $u^+$ is another image in the same category with $u^a$, and $u^-$ is an image in another category that differs from the category of $u^a$. During model training, we can obtain a triplet set consisting of $U^a$, $U^+$, and $U^-$ in each batch, and then randomly select the corresponding samples to form a triple $S = \{u^a, u^+, u^-\}$ as the input of each sub-network. We can then obtain the triplet loss $L_T^m$ for the $m$-th sub-network:

$$L_T^m = Max\{d(f^a, f^+) - d(f^a, f^-) + \alpha, \beta\}. \tag{11}$$

Here, $f^a$, $f^+$, and $f^-$ represent the visual features extracted by the $m$-th sub-network from the images of $u^a, u^+, u^-$, respectively, while $d(\cdot)$ is the Euclidean distance. Moreover, $\alpha$ is a threshold parameter used to distinguish between the positive and negative samples of the anchor samples. $\beta$ is a parameter that is close to 0 without being equal to 0. Triplet loss is used to reduce the distance between the features of $u^a$ and $u^+$ and expand the distance between the features of $u^a$ and $u^-$, as shown in Figure 2. Then, triplet loss can be used to solve the following three situations in HCNNs.

Case I: If $d(f^a, f^+) + \alpha < d(f^a, f^-)$, then $L_T^m = \beta$. This situation shows that the current sub-network can accurately classify these three image samples; thus, there is no need to pay more attention to them in the subsequent sub-networks.

Case II: $d(f^a, f^+) < d(f^a, f^-) < d(f^a, f^+) + \alpha$. This situation shows that high similarity exists among these three image samples, and the current sub-network finds it difficult to distinguish them. This triple $S$ then needs to pass through the subsequent sub-networks with more complex network structures.

Case III: $d(f^a, f^-) < d(f^a, f^+)$. This situation shows that the current sub-network cannot distinguish these image samples, and that their more abstract features need to be extracted by the subsequent sub-networks.

Weighted multi-class joint loss function. HCNNs can progressively classify images and achieve visual feature learning at different levels. In addition, in each batch during model training, a weighted multi-class joint loss function is designed by combining cross-entropy loss with category weights and weighted triplet loss.

$$L^m = L_C^{'(m)} + \gamma L_T^m. \tag{12}$$

Here, $L^m$ is the weighted multi-class joint loss for the $m$-th sub-network. $\gamma$ is a hyperparameter; in this paper, $\gamma = 0.5$.

### *3.3. Model Testing*

To test the proposed model, we need to provide a threshold $H^m$ for each sub-network so as to make the model output the final classification results. When image classification confidence in the $m$-th sub-network is higher than $H^m$, this prediction is reliable; otherwise, the credibility of the image classification results is lower. Generally speaking, the values of $H^m$ can be set larger, which ensures that the difficult-to-identify images can pass through the subsequent sub-networks with more complex network structures. Figure 3 shows the simple process of image classification of HCNNs.

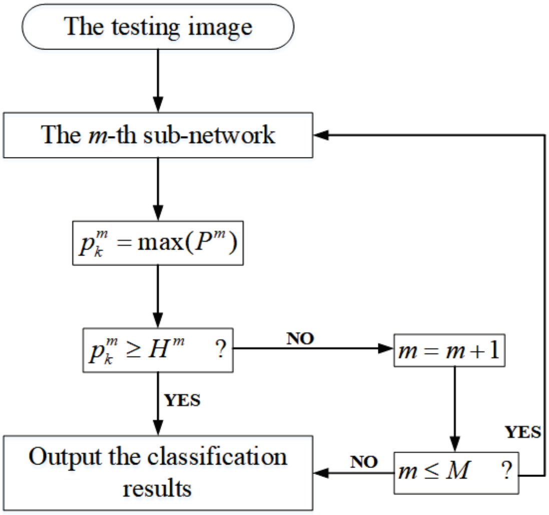


**Figure 3.** Progressive image classification by HCNNs in the test stage.

In more detail, the testing process of HCNNs with $M$ sub-networks can be described as follows.

**Step 1**: The test image is input into the $m$-th sub-network for visual feature learning ($m = 1$ for the first sub-network). The model then outputs the probability distribution of the image classification results $P^m = \{p_1^m, p_2^m, ..., p_{N\_C}^m\}$, where $N\_C$ is the number of image categories.
**Step 2**: A comparison is drawn between the maximum classification probability $p_k^m = Max(P^m)$ and $H^m$.
**Step 3**: If $p_k^m \geq H^m$ or $m = M$, then the model outputs the classification results corresponding to $p_k^m$; otherwise, $m = m + 1$, and return to **Step 1**.

## 4. Experimental Results and Analysis

To verify the effectiveness and superiority of the proposed HCNN, we implement our model on two challenging image classification datasets (our ultrasonic prostate image dataset and the chimpanzee dataset [25]) and two commonly used image classification datasets (CIFAR-10 and CIFAR-100 [24]). Furthermore, we also utilize several related existing DNN models for comparative experimental analysis. In addition, we conduct ablation experiments to verify the influences of different sub-networks.

### *4.1. Image Classification Datasets*

(1) Ultrasonic image dataset of prostate

There have been related works on medical image datasets [51–54], while there are few works for prostate cancer screening. The traditional method of prostate cancer screening usually uses prostate biopsy puncture to obtain the pathological results, which causes great pain for patients. Therefore, we have collected ultrasonic images of prostate, and attempted to design CNN-based models for prostate cancer screening. Our ultrasonic image dataset of prostate has 932 images, which were selected from a number of ultrasonic images according to doctors' experience. We divided these ultrasound images into two categories: the ultrasound images of patients with prostate cancer and those of patients without prostate cancer.

(2) The chimpanzee facial image dataset

The chimpanzee dataset is provided by Loos et al. in [25]. The chimpanzee facial images were captured at Zoo Leipzig in Germany and Taï National Park in Africa. There are large numbers of images with weak or highlight illumination, incomplete facial contours, partial occlusion by branches or leaves, and inconsistent image sizes. This image dataset is therefore very challenging for the image classification task. Table 1 presents the details of the chimpanzee facial images used in this paper. We selected at least five images for each chimpanzee individual from the entire dataset as our test images.

(3) CIFAR-10 and CIFAR-100

CIFAR-10 and CIFAR-100 contain 60,000 images each, where each color image has 32 × 32 pixels. CIFAR-10 comprises 10 categories (aircraft, car, bird, cat, deer, dog, frog, horse, boat, truck). As in previous works, we also use 50,000 images for model training and 10,000 images for testing, and there are no duplicate image samples. In addition, we select 20% of the training images as the validation image dataset. CIFAR-100 consists of the image samples of 100 categories, and we also divide these into 50,000 training images and 10,000 testing images. The training, validation, and testing image sets are allocated according to a ratio of 9:1:2 in the whole CIFAR-100 image dataset (as shown in Table 1).

**Table 1.** Datasets summary.

| Datasets | Category | Train | Validation | Test |
|---|---|---|---|---|
| Ultrasonic image dataset | 2 | 746 | 93 | 93 |
| Chimpanzee facial image dataset | 52 | 1689 | 292 | 540 |
| CIFAR-10 | 10 | 40,000 | 10,000 | 10,000 |
| CIFAR-100 | 100 | 45,000 | 5000 | 10,000 |

### *4.2. Experimental Setup*

The proposed HCNN is an image classification model based on ensemble learning. In this paper, we choose Alexnet [10], VGG16 [11], Inception V3 (I_V3) [49], Mobilenet V2 (M_V2) [50], and Resnet-50 [12] as the basic sub-networks. Of course, in specific tasks, the types of sub-networks can be changed, and we also can add or subtract sub-networks. We further implement several existing deep learning models and ensemble learning models as the comparison models. The experimental results of our model are compared with these basic sub-networks by using the same parameters. We further make comparisons of our model with Adaboost-CNN [38] on CIFAR-10, and with Wide-ResNet 40-2 [55], Wide-ResNet 40-2+CutMix [56], DenseNet-100 [57], and DenseNet-100+CutMix [56] on CIFAR-100. In addition, to verify the gradual improvements achieved by HCNNs, we draw comparisons of HCNNs with the single sub-networks and their different combinations.

For each image dataset, we set the batch size to 10, and the initial learning rate is 0.0001. We set $\gamma$ in Equation (12) to 0.5. The dimensions of $f^a$, $f^+$ and $f^-$ in triplet loss are set to 256 uniformly. All models used in this paper were implemented on three TITAN Xp GPUs.

### *4.3. Experimental Results*

(1) Experimental results on the ultrasonic image dataset

Prostate cancer screening based on ultrasound images is mainly used to distinguish whether the patients have prostate cancer or not according to their prostate ultrasound images. It can be regarded as a binary image classification problem. However, the prostate ultrasound images are fairly complex, and the lesions are not obvious, so it is difficult for professional doctors to diagnose prostate diseases only using ultrasound images. Therefore, there are challenges for the automatic screening of prostate cancer utilizing computer vision technologies from ultrasound images. In this paper, we try to use several DNN models to perform the binary image classification task, and the experimental results are shown in Table 2. We can obtain the following points from the experimental results. First, all the models used in this paper fail to achieve perfect image classification performance, and the highest recognition accuracy is lower than 85%, which shows that there is difficulty in recognizing prostate cancer. Second, Resnet50 achieves better performance among the single deep network models. It has 4.31% higher accuracy than VGG16, and 5.65% higher than Inception V3. Third, the models combing multiple networks have increasing accuracies; for example, "Alexnet+VGG16+Inception V3+Mobilenet V2" has 2.83% higher accuracy than VGG16. Among these methods, the HCNN with five deep networks achieves the best performance, and it has 2.68% higher accuracy than Resnet50. Figure 4 shows the graph of Table 2; we can see that HCNN achieves obvious advantages over other models in most evaluation indicators. In addition, the performance would be further improved with the improvement or addition of the sub-networks.

**Table 2.** The ablation analysis on the ultrasonic image dataset.

| Models | ACC | F1 Score | Recall | Precision |
|---|---|---|---|---|
| Alexnet [10] | 0.6989 | 0.7200 | 0.8571 | 0.6207 |
| VGG16 [11] | 0.7634 | 0.7381 | 0.7381 | 0.7381 |
| Inception V3 [49] | 0.7500 | 0.7164 | 0.7273 | 0.7059 |
| Mobilenet V2 [50] | 0.6989 | 0.6499 | 0.6190 | 0.6842 |
| Resnet50 [12] | 0.8065 | 0.7805 | 0.7619 | 0.8000 |
| Alexnet+VGG16 | 0.7361 | 0.7077 | 0.7188 | 0.6970 |
| Alexnet+VGG16+Inception V3 | 0.7639 | 0.7385 | 0.7500 | 0.7273 |
| Alexnet+VGG16+ Inception V3+Mobilenet V2 | 0.7917 | 0.7693 | 0.7813 | 0.7576 |
| HCNN | **0.8333** | **0.8125** | **0.8125** | **0.8125** |

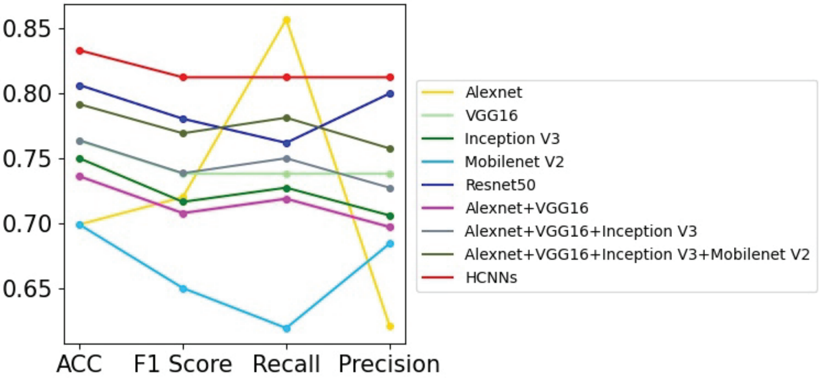


**Figure 4.** The curve graph of Table 2.

(2) Experimental results on the chimpanzee facial image dataset

There are a large number of challenging chimpanzee facial images in the chimpanzee facial image dataset [25]; it is therefore difficult for models to achieve good image classification performance. In this paper, we implement nine different models on this dataset; the experimental results are shown in Table 3, where we can see that these models, which perform well on some public image classification databases, do not perform well on this chimpanzee facial image dataset. The model with the single network that works best is Resnet50, with 0.7336% accuracy, while the HCNN achieves the highest image classification accuracy of 74.55%. Figure 5 shows the curves of these models' performance in related evaluation indicators, and it can be seen that the HCNN has general advantages over models with single neural networks and other models with multiple sub-networks in terms of F1 score, recall, and precision, which clearly demonstrates its effectiveness and superiority.

**Table 3.** The ablation analysis on the chimpanzee facial image dataset.

| Models | ACC | F1 Score | Recall | Precision |
|---|---|---|---|---|
| Alexnet [10] | 0.5532 | 0.5470 | 0.5428 | 0.5512 |
| VGG16 [11] | 0.6885 | 0.6836 | 0.6818 | 0.6854 |
| Inception V3 [49] | 0.7008 | 0.6976 | 0.6956 | 0.6996 |
| Mobilenet V2 [50] | 0.5737 | 0.5699 | 0.5701 | 0.5698 |
| Resnet50 [12] | 0.7336 | 0.7327 | 0.7321 | 0.7334 |
| Alexnet+VGG16 | 0.7023 | 0.7010 | 0.6998 | 0.7023 |
| Alexnet+VGG16+Inception V3 | 0.7234 | 0.7200 | 0.7199 | 0.7201 |
| Alexnet+VGG16+Inception V3+Mobilenet V2 | 0.7349 | 0.7316 | 0.7288 | 0.7344 |
| HCNN | **0.7455** | **0.7435** | **0.7451** | **0.7419** |

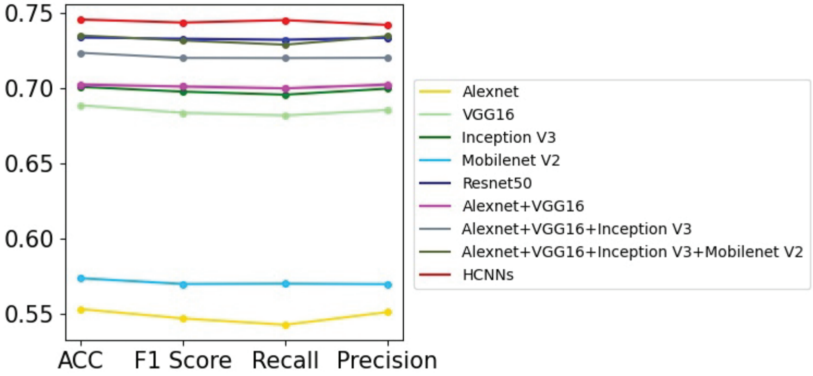


**Figure 5.** The curve graph of Table 3.

(3) Experimental results on CIFAR-10

Table 4 presents the experimental results of nine different models on CIFAR-10; these models include the sub-networks in HCNNs and their different combinations. For each model, we carried out 20 epochs of model training. From the results shown in Table 4, we can see that Resnet50 [12] achieves the best performance among all models with a single network. Furthermore, the ensemble learning models with different sub-networks achieve general improvements over the corresponding single-network models, which proves the effectiveness of ensemble learning models. Therefore, HCNN achieves the final best performance, with test accuracy of 92.26% on CIFAR-10, and 1.46–13.45% higher classification accuracy than the other five basic sub-networks. In addition, HCNN also has advantages in terms of F1 score, recall, and precision. In Figure 6, the performance difference among these models can be illustrated more clearly. Although these models achieved better results on CIFAR-10 than on the ultrasonic image dataset and chimpanzee facial image dataset, the overall trends of the models' performance are similar. The reasons may be that the weights of the training samples for each sub-network in HCNNs are updated according to the classification results of the previous sub-network. In this way, different sub-networks can learn the specific features from the images, and present various degrees of difficulty to the various models that attempt to accurately classify them. Therefore, HCNNs can progressively learn the visual features at different levels and gradually improve the image classification performance.

**Table 4.** The ablation analysis on the CIFAR-10 dataset.

| Models | ACC | F1 Score | Recall | Precision |
|---|---|---|---|---|
| Alexnet [10] | 0.7881 | 0.7881 | 0.7894 | 0.7868 |
| VGG16 [11] | 0.8860 | 0.8810 | 0.8818 | 0.8802 |
| Inception V3 [49] | 0.8922 | 0.8921 | 0.8928 | 0.8914 |
| Mobilenet V2 [50] | 0.8704 | 0.8639 | 0.8647 | 0.8631 |
| Resnet50 [12] | 0.9080 | 0.9008 | 0.9011 | 0.9005 |
| Alexnet+VGG16 | 0.8877 | 0.8878 | 0.8880 | 0.8877 |
| Alexnet+VGG16+Inception V3 | 0.9023 | 0.9086 | 0.9087 | 0.9085 |
| Alexnet+VGG16+Inception V3+Mobilenet V2 | 0.9104 | 0.9106 | 0.9107 | 0.9105 |
| HCNN | **0.9226** | **0.9221** | **0.9222** | **0.9221** |

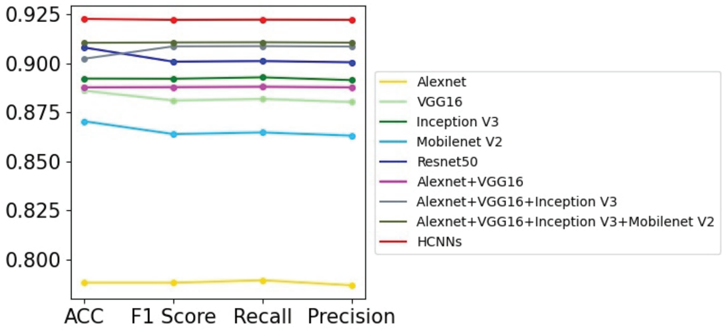


**Figure 6.** The curve graph of Table 4.

AdaBoost-CNN [38], proposed by Taherkhani et al., is also an ensemble learning model based on DNNs. Its test accuracy on CIFAR-10 reaches 81.40%, as shown in Table 5. By contrast, our HCNN has 10.86% higher accuracy. Adaboost-CNN creates a classification model with better performance by combining several simple convolutional sub-networks. However, multiple sub-networks in Adaboost-CNN use the same network structure, and each sub-network is only fine-tuned on the parameters of its previous sub-network. Therefore, it is difficult for the model to learn the specific abstract visual features from the images; this may be the reason for the limited performance of Adaboost-CNN.

**Table 5.** The experimental analysis of Adaboost-CNN and HCNN on the CIFAR-10 dataset.

| Models | Acc |
|---|---|
| Adaboost-CNN [38] | **0.8140** |
| HCNN | **0.9226** |

(4) Experimental results on CIFAR-100

CIFAR-100 contains more image categories than CIFAR-10 and is less able to achieve higher image classification accuracy for classification models. Fourteen different models are implemented in CIFAR-100, and the experimental results are shown in Tables 6 and 7 and Figure 7. From the test results shown in Table 6, it can be seen that the models combining different sub-networks achieve better performance than models with a single neural network, which is similar to Table 4. The HCNN, with a test accuracy of 78.47%, achieves accuracy that is 9.46% higher than that of Mobilenet V2 and 1.72% higher than that of Inception V3, which represents the best performance among the single neural network models.

Moreover, as shown in Table 7, HCNN achieves similar performance to DenseNet-100+CutMix [56], but is better than other existing network models with complex network structures. During the image classification process, each image (regardless of whether it is easy or difficult for the model to accurately classify) needs to go through these complex network models to extract visual features. In HCNNs, however, different images will pass through different levels of sub-networks, and the model will learn specific visual features from images at different levels. Therefore, HCNN achieves better effectiveness and efficiency for image classification.

**Table 6.** The ablation analysis on the CIFAR-100 dataset.

| Models | ACC | F1 Score | Recall | Precision |
|---|---|---|---|---|
| Alexnet [10] | 0.5347 | 0.5326 | 0.5329 | 0.5323 |
| VGG16 [11] | 0.6556 | 0.6548 | 0.6556 | 0.6540 |
| Inception V3 [49] | 0.7675 | 0.7686 | 0.7690 | 0.7682 |
| Mobilenet V2 [50] | 0.6601 | 0.6634 | 0.6645 | 0.6623 |
| Resnet50 [12] | 0.6031 | 0.6033 | 0.6034 | 0.6032 |
| Alexnet+VGG16 | 0.6623 | 0.6640 | 0.6646 | 0.6635 |
| Alexnet+VGG16+Inception V3 | 0.7742 | 0.7767 | 0.7778 | 0.7757 |
| Alexnet+VGG16+Inception V3+Mobilenet V2 | 0.7798 | 0.7740 | 0.7746 | 0.7735 |
| HCNN | **0.7847** | **0.7846** | **0.7844** | **0.7848** |

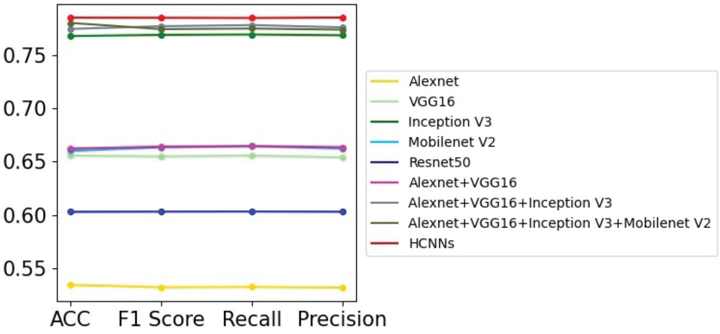


**Figure 7.** The curve graph of Table 6.

**Table 7.** The experimental results of Adaboost-CNN and HCNN on the CIFAR-100 dataset.

| Models | Acc |
|---|---|
| Wide-ResNet 40-2 [55] | 0.7473 |
| Wide-ResNet 40-2+CutMix [56] | 0.7821 |
| DenseNet-100 [57] | 0.7773 |
| DenseNet-100+CutMix [56] | **0.7855** |
| HCNN | **0.7847** |

## 5. Conclusions

At present, all the image classification models treat the images equally. However, there are meaningful differences between images, so different images should be treated differently by various models, which would comply with the basic mechanism of human cognition. Therefore, we propose HCNNs, which classify different images by different numbers of sub-networks. In HCNNs, the easy-to-identify images are recognized by simple sub-networks and output the results directly, while images that are more difficult to identify may need to go through multiple complex sub-networks to extract their more abstract visual features. Through this image classification mechanism, HCNNs achieve better image classification performance compared with existing single-network models and Adaboost CNN with its multiple simple sub-networks. In addition, the HCNN has better scalability and variability; that is, the number of sub-networks can be increased or decreased, and the types of sub-networks can be changed according to the specific visual tasks involved. Therefore, in the future, more detailed models similar to HCNNs may be constructed based on the complexity of the image classification task, which would gradually become closer to the basic mechanism of human cognition, and the models will have higher recognition accuracy and efficiency.

**Author Contributions:** C.L. designed the research and wrote the manuscript. F.M. contributed to the improvement of the ideas and to the revision of the manuscript. G.G. carried out the data collection and research experiments. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Natural Science Foundation of China under grant agreements Nos. 61973250, 61973249, 62073218, 61802335, 61902313, 61902296, 61971349; Shaanxi Province Science Fund for Distinguished Young Scholars: 2018JC-016; Shaanxi Provincial Department of Education serves local scientific research: 19JC038; and the Key Research and Development Program of Shaanxi: 2021GY-077, 2020ZDLGY04-07, 2021ZDLGY02-06, 2019GY-012.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The images in Ultrasonic image dataset of prostate are collected from Department of Ultrasound, Ruijin Hospital, Shanghai Jiao Tong University School of Medicine, Shanghai, China, and these used images have no personal information of patients. Chimpanzee facial image dataset, CIFAR-10 and CIFAR-100 are public image datasets.

**Conflicts of Interest:** All the authors declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Nie, L.; Zhang, L.; Meng, L.; Song, X.; Chang, X.; Li, X. Modeling disease progression via multisource multitask learners: A case study with Alzheimer's disease. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 1508–1519. [CrossRef]
2. Luo, M.; Chang, X.; Nie, L.; Yang, Y.; Hauptmann, A.G.; Zheng, Q. An adaptive semisupervised feature analysis for video semantic recognition. *IEEE Trans. Cybern.* **2017**, *48*, 648–660. [CrossRef]
3. Wang, S.; Chang, X.; Li, X.; Long, G.; Yao, L.; Sheng, Q.Z. Diagnosis code assignment using sparsity-based disease correlation embedding. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 3191–3202. [CrossRef]
4. Qi, L.; Tang, W.; Zhou, L.; Huang, Y.; Zhao, S.; Liu, L.; Li, M.; Zhang, L.; Feng, S.; Hou, D.; et al. Long-term follow-up of persistent pulmonary pure ground-glass nodules with deep learning–assisted nodule segmentation. *Eur. Radiol.* **2020**, *30*, 744–755. [CrossRef] [PubMed]
5. Munir, K.; Elahi, H.; Ayub, A.; Frezza, F.; Rizzi, A. Cancer Diagnosis Using Deep Learning: A Bibliographic Review. *Cancers* **2019**, *11*, 1235. [CrossRef] [PubMed]
6. Liu, B.; Chi, W.; Li, X.; Li, P.; Liang, W.; Liu, H.; Wang, W.; He, J. Evolving the pulmonary nodules diagnosis from classical approaches to deep learning-aided decision support: Three decades' development course and future prospect. *J. Cancer Res. Clin. Oncol. Vol.* **2020**, *146*, 153–185. [CrossRef] [PubMed]
7. Cheng, Z.; Chang, X.; Zhu, L.; Kanjirathinkal, R.C.; Kankanhalli, M. MMALFM: Explainable recommendation by leveraging reviews and images. *ACM Trans. Inf. Syst. (TOIS)* **2019**, *37*, 1–28. [CrossRef]
8. Li, Z.; Yao, L.; Chang, X.; Zhan, K.; Sun, J.; Zhang, H. Zero-shot event detection via event-adaptive concept relevance mining. *Pattern Recognit.* **2019**, *88*, 595–603. [CrossRef]
9. Yu, E.; Sun, J.; Li, J.; Chang, X.; Han, X.H.; Hauptmann, A.G. Adaptive semi-supervised feature selection for cross-modal retrieval. *IEEE Trans. Multimed.* **2018**, *21*, 1276–1288. [CrossRef]
10. Krizhevsky, A.; Sutskever, L.; Hinton, G.E. Imagenet classification withdeep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105.
11. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recog-nition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
13. Forres, I.; Matt, M.; Serge, K.; Ross, G.; Trevor, K.; Kurt, K. Densenet: Im-plementing efficient convnet descriptor pyramids. *arXiv* **2014**, arXiv:1404.1869.
14. Qiang, L.; Xuyu, X.; Jiaohua, Q.; Yun, T.; Yuanjing, T.J.L. Cover-less steganography based on image retrieval of densenet features and dwtsequence mapping. *Knowl.-Based Syst.* **2020**, *192*, 105375.
15. Zhang, D.; Yao, L.; Chen, K.; Wang, S.; Chang, X.; Liu, Y. Making sense of spatio-temporal preserving representations for EEG-based human intention recognition. *IEEE Trans. Cybern.* **2019**, *50*, 3033–3044. [CrossRef] [PubMed]
16. Nie, L.; Zhang, L.; Yan, Y.; Chang, X.; Liu, M.; Shaoling, L. Multiview physician-specific attributes fusion for health seeking. *IEEE Trans. Cybern.* **2016**, *47*, 3680–3691. [CrossRef]
17. Ren, P.; Xiao, Y.; Chang, X.; Huang, P.Y.; Li, Z.; Chen, X.; Wang, X. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–34. [CrossRef]

18. Li, Z.; Nie, F.; Chang, X.; Yang, Y. Beyond trace ratio: Weighted harmonic mean of trace ratios for multiclass discriminant analysis. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2100–2110. [CrossRef]
19. Yuan, D.; Chang, X.; Huang, P.Y.; Liu, Q.; He, Z. Self-supervised deep correlation tracking. *IEEE Trans. Image Process.* **2020**, *30*, 976–985. [CrossRef]
20. Ma, Z.; Chang, X.; Yang, Y.; Sebe, N.; Hauptmann, A.G. The many shades of negativity. *IEEE Trans. Multimed.* **2017**, *19*, 1558–1568. [CrossRef]
21. Li, Z.; Nie, F.; Chang, X.; Yang, Y.; Zhang, C.; Sebe, N. Dynamic affinity graph construction for spectral clustering using multiple features. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 6323–6332. [CrossRef] [PubMed]
22. Yan, C.; Zheng, Q.; Chang, X.; Luo, M.; Yeh, C.H.; Hauptman, A.G. Semantics-preserving graph propagation for zero-shot object detection. *IEEE Trans. Image Process.* **2020**, *29*, 8163–8176. [CrossRef] [PubMed]
23. Ren, P.; Xiao, Y.; Chang, X.; Huang, P.Y.; Li, Z.; Gupta, B.; Wang, X. A Survey of Deep Active Learning. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–40. [CrossRef]
24. Krizhevsky, A.; Hinton, G. Learning multiple layers of features fromtiny images. *Tech Rep.* **2009**, *7*, 1–60.
25. Loos, A.; Ernst, A. An automated chimpanzee identification system using face detection and recognition. *EURASIP J. Image Video Process.* **2013**, *2013*, 49. [CrossRef]
26. Khan, S.; Nazir, S.; Garcia-Magarino, I.; Hussain, A. Deep learning-base urban big data fusion in smart cities: Towards traffic monitoring and flow-preserving fusion. *Comput. Electr. Eng.* **2021**, *89*, 106906. [CrossRef]
27. Orozco, M.C.E.; Rebong, C.B. Vehicular detection and classification forintelligent transportation system: A deep learning approach using faster490r-cnn model. *Int. J. Simul. Syst.* **2019**, *180*, 36551.
28. Zhenghao, X.; Niu, Y.; Chen, J.; Kan, X.; Liu, H. Facial expression recognition of industrial internet of things by parallel neural networks combining texture features. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2784–2793.
29. Hossain, M.S.; Muhammad, G.; Amin, S.U. Improving consumer satisfac-tion in smart cities using edge computing and caching: A case study ofdate fruits classification. *Future Gener. Comput. Syst.* **2018**, *88*, 333–341. [CrossRef]
30. Gören, S.; Óncevarlk, D.F.; Yldz, K.D.; Hakyemez, T.Z. On-street parking500spot detection for smart cities. In Proceedings of the IEEE International Smart CitiesConference (ISC2), Casablanca, Morocco, 14–17 October 2019; pp. 292–295.
31. Yao, H.; Gao, P.; Wang, J.; Zhang, P.; Jiang, C.; Han, Z. Capsule networkassisted iot traffic classification mechanism for smart cities. *IEEE Internet Things J.* **2019**, *6*, 7515–7525. [CrossRef]
32. Hassan, A.; Liu, F.; Wang, F.; Wang, Y. Secure image classification withdeep neural networks for iot applications. *J. Ambient Intell. Humaniz. Comput.* **2020**, *12*, 8319–8337. [CrossRef]
33. Vasan, D.; Alazab, M.; Wassan, S.; Naeem, H.; Safaei, B.; Zheng, Q. Imcfn:Image-based malware classification using fine-tuned convolutional neural network architecture. *Comput. Netw.* **2020**, *171*, 107138. [CrossRef]
34. Ciregan, D.; Meier, U.; Schmidhuber, J. Multi-column deep neural networksfor image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3642–3649.
35. Frazao, X.; Alexandre, L.A. Weighted convolutional neural network ensemble. In *Iberoamerican Congress on Pattern Recognition*; Springer: Cham, Switzerland, 2014; pp. 674–681.
36. Tajbakhsh, N.; Gurudu, S.R.; Liang, J. Automatic polyp detection incolonoscopy videos using an ensemble of convolutional neural networks. In Proceedings of the 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI), Brooklyn, NY, USA, 16–19 April 2015; pp: 79–83.
37. Ijjina, E.P.; Mohan, C.K. Hybrid deep neural network model for humanaction recognition. *Appl. Soft Comput.* **2016**, *46*, 936–952. [CrossRef]
38. Taherkhani, A.; Cosma, G.; McGinnity, T.M. Adaboost-cnn: An adaptiveboosting algorithm for convolutional neural networks to classify multi-classimbalanced datasets using transfer learning. *Neurocomputing* **2020**, *404*, 351–366. [CrossRef]
39. Movshovitz-Attias, Y.; Toshev, A.; Leung, T.K.; Ioffe, S.; Singh, S. No fussdistance metric learning using proxies. In Proceedings of the IEEE International Conferenceon Computer Vision, Venice, Italy, 22–29 October 2017; pp. 360–368.
40. Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. Sphereface: Deep hy-persphere embedding for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 212–220.
41. Chen, W.; Chen, X.; Zhang, J.; Huang, K. Beyond triplet loss: A deepquadruplet network for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 403–412.
42. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding forface recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
43. Wang, X.; Han, X.; Huang, W.; Dong, D.; Scott, M.R. Multi-similarity loss with general pair weighting for deep metric learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5022–5030.
44. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp: 499–515.
45. Wang, J.; Zhou, F.; Wen, S.; Liu, X.; Lin, Y. Deep metric learning withangular loss. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2593–2601.

46. Jiang, L.; Meng, D.; Yu, S.-I.; Lan, Z.; Shan, S.; Hauptmann, A. Self-paced learning with diversity. *Adv. Neural Inf. Process. Syst*. **2014**, *27*, 2078–2086.
47. Chang, X.; Yu, Y.L.; Yang, Y.; Xing, E.P. Semantic pooling for complex event analysis in untrimmed videos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1617–1632. [CrossRef]
48. Yan, C.; Chang, X.; Li, Z.; Guan, W.; Ge, Z.; Zhu, L.; Zheng, Q. ZeroNAS: Differentiable Generative Adversarial Networks Search for Zero-Shot Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef] [PubMed]
49. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking theinception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
50. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2:Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
51. Munir, K.; Frezza, F.; Rizzi, A. Deep Learning for Brain Tumor Segmentation. In *Deep Learning for Cancer Diagnosis*; Springer: Singapore, 2020; pp. 189–201.
52. Munir, K.; Elahi, H.; Farooq, M.U.; Ahmed, S.; Frezza, F.; Rizzi, A. Detection and screening of COVID-19 through chest computed tomography radiographs using deep neural networks. In *Data Science for COVID-19*; Academic Press: Cambridge, MA, USA, 2021; pp. 63–73.
53. Munir, K.; Frezza, F.; Rizzi, A. Brain Tumor Segmentation Using 2D-UNET Convolutional Neural Network. In *Deep Learning for Cancer Diagnosis*; Springer: Singapore, 2020; pp. 239–248.
54. Fakoor, R.; Ladhak, F.; Nazi, A.; Huber, M. Using deep learning to enhance cancer diagnosis and classification. In Proceedings of the 30th International Conference on Machine Learning (ICML 2013), Atlanta, GA, USA, 16–21 June 2013.
55. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146.
56. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regulariza-tion strategy to train strong classifiers with localizable features. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6023–6032.
57. Huang, G.; Liu, Z.; Maaten, L.V.D.; Weinberger, K.Q. Densely con-nected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.

*electronics*

*Article*

# Lesion Segmentation Framework Based on Convolutional Neural Networks with Dual Attention Mechanism

**Fei Xie [1,2,†], Panpan Zhang [3,†], Tao Jiang [4], Jiao She [3], Xuemin Shen [5,*], Pengfei Xu [3,*], Wei Zhao [6], Gang Gao [7] and Ziyu Guan [6]**

[1] School of AOAIR, Xidian University, Xi'an 710075, China; fxie@xidian.edu.cn
[2] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[3] School of Information Science and Technology, Northwest University, Xi'an 710069, China; 201932137@stumail.nwu.edu.cn (P.Z.); shejiao@stumail.nwu.edu.cn (J.S.)
[4] Zhejiang Provincial Seaport, Ningbo 315040, China; jiangt@nbport.com.cn
[5] Department of Oral Mucosal Diseases, Shanghai Ninth People's Hospital, College of Stomatology, Shanghai Jiao Tong University School of Medicine, Shanghai 20025, China
[6] School of Computer Science and Technology, Xidian University, Xi'an 710075, China; ywzhao@mail.xidian.edu.cn (W.Z.); zyguan@xidian.edu.cn (Z.G.)
[7] Shaanxi Great Wisdom Medical Care Technologies Co., Ltd., Xi'an 710075, China; xiefei@nwpu.edu.cn
* Correspondence: SHENXM1327@sh9hospital.org.cn (X.S.); pfxu@nwu.edu.cn (P.X.)
† They are the Co-first author.

**Abstract:** Computational intelligence has been widely used in medical information processing. The deep learning methods, especially, have many successful applications in medical image analysis. In this paper, we proposed an end-to-end medical lesion segmentation framework based on convolutional neural networks with a dual attention mechanism, which integrates both fully and weakly supervised segmentation. The weakly supervised segmentation module achieves accurate lesion segmentation by using bounding-box labels of lesion areas, which solves the problem of the high cost of pixel-level labels with lesions in the medical images. In addition, a dual attention mechanism is introduced to enhance the network's ability for visual feature learning. The dual attention mechanism (channel and spatial attention) can help the network pay attention to feature extraction from important regions. Compared with the current mainstream method of weakly supervised segmentation using pseudo labels, it can greatly reduce the gaps between ground-truth labels and pseudo labels. The final experimental results show that our proposed framework achieved more competitive performances on oral lesion dataset, and our framework further extended to dermatological lesion segmentation.

**Keywords:** medical image segmentation; computational intelligence; convolutional neural networks; weakly supervised segmentation; attention mechanism

**Citation:** Xie, F.; Zhang, P.; Jiang, T.; She, J.; Shen, X.; Xu, P.; Zhao, W.; Gao, G.; Guan, Z. Lesion Segmentation Framework Based on Convolutional Neural Networks with Dual Attention Mechanism. *Electronics* **2021**, *10*, 3103. https://doi.org/10.3390/electronics10243103

Academic Editor: Daniel Gutiérrez Reina

Received: 31 October 2021
Accepted: 7 December 2021
Published: 13 December 2021

## 1. Introduction

With the rapid development of computer vision, especially the significant improvement of the representation ability of convolutional neural networks [1,2], image segmentation has achieved good performances and laid a solid foundation for the application of medical image segmentation. Medical images segmentation as an important and difficult task of computer-aided diagnosis, is the key to further obtain diagnostic information. Traditional object location in medical images requires professional doctors to manually identify, which is not only time-consuming and labor-intensive but also vulnerable to subjective factors. While the lesion segmentation results obtained by deep learning methods are now becoming a promising method. However, compared with ordinary images, clinical diagnosis invokes higher requirements for the accuracy of the segmentation results of medical images. In addition, the high variability, the complex morphological structure, the ambiguity and the scarce labels of lesions in medical images pose great challenges to medical image segmentation [3].

Recently, lesion segmentation methods based on deep convolutional neural networks have been widely applied to medical image segmentation. Encoder-Decoder, FCNs [4] (Fully Convolutional Networks for Semantic Segmentation) and the methods based on extended convolutional neural network have become the mainstream segmentation methods. For example, U-Net designed in [5] an "U-shaped" network, and symmetric expansion paths are added to enhance the positioning representation capability of the network. U-Net is superior to the previous methods in terms of the amount of data required, the efficiency and accuracy of methods. Since then, more and more variants of U-Net [6–11] are proposed to enhance the network presentation capabilities, the transmission and fusion of feature information within and between layers to further improve the segmentation accuracy. U-net and its variants perform well in medical images such as CT(Computed Tomography) and MRI(Magnetic Resonance Imaging). On the one hand, the CT and MRI images are mostly single-channel grayscale images, with simple semantics and relatively fixed structures. On the other hand, the U-Net network has fewer parameters, and the skip connection of U-net plays an important role. The skip connection can make the feature graph of the corresponding position of the encoder fuse on the channel in the up-sampling process of each level of the network. Through the fusion of low-level features and high-level features, the network can retain more high-resolution details contained in high-level feature images, thus improving the accuracy of image segmentation, so as it is not easy to overfit for relatively small medical datasets. Therefore, when there are relatively small medical datasets, this U-NET model is preferred to avoid overfitting.

There are also some medical image datasets consisting of visible images, such as our oral leukoplakia dataset and the ISIC2018 [12–14] used in this paper. Different from radiographic images such as CT and MRI images, this type of medical image taken by conventional visible light cameras have larger size, and relatively richer semantic information, while they also have challenges in terms of object segmentation. As shown in Figure 1, the same category of objects has some differences in visual features, while the features of different categories of objects have similarities. The texture, color, shape, and size of lesions in the images varies, and the boundary of lesions is blurred. In addition, all of the artifacts during image capturing, light intensity and reflections, bubbles, hair occlusion, background boards, and so forth, bring many difficulties to the segmentation task. Specifically, for the oral leukoplakia dataset built in this paper, the difficulties of leukoplakia lesion segmentation mainly lie in the morphology diversity of lesion, including granular, crumpled, warty, and so forth. In addition, the differences in the size of lesions, the blur boundaries between the lesions and their surrounding tissues, and the changeable locations of lesions, and so forth, will also increase the difficulty in the segmentation of leukoplakia. At present, there are few related works to oral lesion segmentation. Camalan et al. [15] developed a image classification method to identify the "suspicious" oral dysplasia or "normal" oral images through transfer learning on Inception-ResNet-V2. Jubair et al. [16] proposed a method to predict oral cancer from oral images using a lightweight transfer learning model. These methods are designed for the image classification task, while our method is performed for segmenting the oral lesion, which is the image segmentation task with more complexity. Figure 1 shows some examples of segmentation results of oral leukoplakia and skin disease lesions. All of these complex medical images pose big challenges for lesion segmentation, since the category filling rate loss in traditional image segmentation models often loses its effects of medical image segmentation.

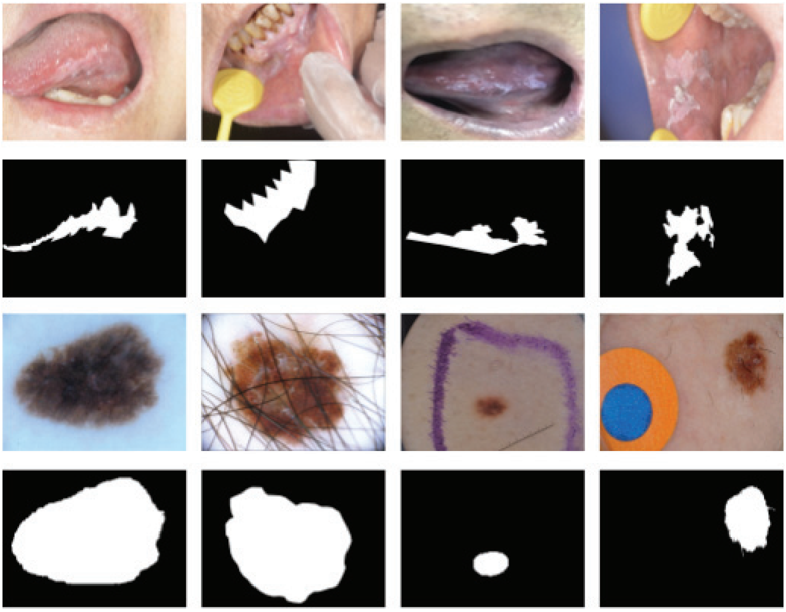


**Figure 1.** Some samples of lesion segmentation (The first two rows are oral leukoplakia dataset, and the last two rows are ISIC 2018 dataset).

For some types of medical images with extremely complex morphological and multi-scale features, traditional U-Net can not deal with the multi-scale features of the objects well, which makes it difficult for U-Net to extract the effective visual features of small objects. This results from the fact that the models have significant performance degradation. In order to solve these problems, it is particularly important to improve the network's visual feature capturing capabilities for medical images. Fortunately, many attempts have been made, and introducing the attention mechanism into different deep networks is a feasible direction. Compared with the U-Net, the structure of Mask R-CNN is more complex, especially the FPN backbone network. This network can adapt to multi-scale changes of lesions and extract effective regional features. Therefore, in this paper, we introduce a dual attention mechanism into Mask R-CNN [17–20], and propose a network to extract the effective visual features of lesions. The dual attention mechanism can help the network pay attention to feature extraction from important regions, which can improve the representation ability of the convolutional networks for lesion areas. The experimental results show that the models with the dual attention mechanism have the optimal segmentation boundaries, and fewer missed or false segmentation areas. Although it is possible to obtain better segmentation results through fully supervised learning, it requires fine pixel-level labels of the objects in images. Therefore, a professional pathologist is required to give labels for medical images, which can consume some economic and time and greatly limits the practical application of intelligent assistance systems. To solve this problem, the segmentation methods based on weakly supervised learning can use image-level and box-level coarse-grained labels to train a pixel-level fine segmentation network. In this paper, we make full use of our network with a dual attention mechanism, and integrate a weakly supervised segmentation branch. This improvement achieves the weakly supervised lesion segmentation, which has much less cost in image labeling. Furthermore, this segmentation framework can also be applied to radiomics lesion segmentation in the future.

Main contributions can be concluded as follows:

1. In this paper, the researchers construct an end-to-end medical lesion segmentation framework, which has both fully supervised segmentation and weakly supervised segmentation branches. If pixel-level labels are used for the images, the fully supervised segmentation branch can be used for lesion segmentation. In the process of experiments if we only have box-level labels similar to the labels for object detection, the researchers can use the weakly supervised segmentation branch to achieve accurate lesion segmentation with comparable results to those obtained by the fully supervised segmentation methods.
2. To solve the problem of inaccurate segmentation of lesion boundaries in the lesion segmentation task, the researchers introduce the CBAM [21] attention mechanism into the Mask R-CNN to help the network pay attention to fine-gain feature learning from the regions of interest. This improvement will be beneficial for the segmentation results, especially for the segmented lesion boundaries.

This paper is organized as follows. In Section 2, we first state some classical methods of fully and weakly supervised image segmentation (Sections 2.1 and 2.2), as well as the basic principles of attention mechanisms (Section 2.3). Then, the proposed image segmentation framework is described in Section 3, and we specifically give a statement for weakly supervision segmentation in Section 3.2. The validation experiments and analysis of the model are described in Section 4; here, we carried out several related methods and our improved method on public skin image datasets and our own oral leukoplakia image dataset. While the final conclusion is given in Section 5.

## 2. Materials and Methods

### *2.1. Fully Supervised Segmentation Methods*

Fully supervised segmentation is divided into semantic segmentation and instance segmentation. In terms of semantic segmentation, since FCN [22] first introduced full convolutional neural networks into segmentation, a series of improved or redesigned segmentation methods [23–29] following this paradigm have achieved good results. Currently, models based on the encoder-decoder structure have gradually become the mainstream segmentation framework such as SegNet [30,31], U-Net [5] and RefineNet [32]. The main reason is that this model can extract long-distance semantic information. In addition, ParseNet [33], DeepLabv2 [34], PSPNet [35] and other models based on spatial pyramid pooling [36] to capture long-distance contextual semantic features are becoming popular as well. In addition, among the methods mentioned above, Refine Net is a good network model. This model is a multipath optimized network for high resolution semantic segmentation. It makes perfect use of all available information in the downsampling process to achieve high resolution prediction of long-distance residual connections. Moreover, a network structure for generating high rate segmentation graph is provided by combining high level semantic features with low level features. This feature makes it suitable for multi-class semantic segmentation tasks. Compared with semantic segmentation, instance segmentation also needs to distinguish different instances of the same class of targets on the basis of semantic segmentation. Many methods [37–42] incorporating the region proposal network [43] (RPN) have achieved satisfying results. These methods first obtain the detection box of the target and then use another segmentation branches to segment the instance.

Among the new methods, Mask-RCNN [18] adds a mask branch based on faster R-CNN [44]. This branch changes ROI pooling to ROI alignment, so as to obtain pixel-level mask prediction. It also has the functions of object detection and instance segmentation. This method has become a general framework.

### *2.2. Weakly Supervised Segmentation Methods*

The fully-supervised segmentation model can segment accurate results after training with a large number of pixel-wise labels, but it is extremely expensive to obtain pixel-wise labels. To address this issue, different levels of weakly supervised labels are adopted to solve the problem of manually labeling large amounts of data, such as image-level [39,45–50], scribbles [51–53], and point labels [54]. Because these weakly supervised labels provide limited prior information, it is difficult to produce satisfactory results for the complex medical images. In this article, we focus on using bounding box-level labels to balance labeling cost and segmentation accuracy. Previous box-level weakly supervised segmentation methods usually need to manually generate weakly supervised pseudo labels, and then use pseudo labels for training on fully supervised methods. Specifically, it can be divided into three stages: the first stage uses GrabCut [55] or MCG [56] to generate pixel-level pseudo labels, the second stage uses the generated pseudo labels as ground-truth to train the segmentation model, and the third stage uses an iterative algorithm or conditional random fields (CRFs) [2,57–60] to optimize and post-process the segmentation results. Therefore, it is difficult to solve the gap between the ground-truth and the pseudo label, and the final segmentation result will be significantly worse than the effect of the pseudo label. Compared with previous work, our weakly supervised segmentation method is different in the following respects. Firstly, our model does not require manual generation of pseudo labels. Secondly, we first use box labels to train the target detection model, and then use GrabCut [55] to separate the foreground and background regions of the detected target region in the inference stage. Thirdly, in terms of final mask optimization, we abandoned the use of CRFs [2,57–60], and instead adopted the faster ConvCRF [20,61–63] for post-processing.

### *2.3. Attention Mechanism*

Human vision can quickly scan the global image to obtain the target areas that need to be focused on. The attention mechanism in deep learning is similar to the human visual attention mechanism, and the goal is to select the critical information in the current task. By adjusting the feature map, Wang proposed a residual attention network [64,65], which not only performs better but also is robust to noisy input. Oktay proposed the Attention-Unet [6,66] to suppress the information of unimportant regions, which is better in segmentation. Hu [67–69] proposed the Squeeze-and-Excitation module based on the relationship between channels. It only uses global average pooling to calculate channel attention. However, as shown in spatial attention[70], it also plays an important role in convolutional networks. It will tell the network “where” to focus. Since then, the application of channel attention and spatial attention have become a consensus. DANet [71] introduces the idea of self-attention, which can better capture features through a long-range context. CCNet [72,73] skillfully uses the criss-cross idea, which greatly reduces the amount of calculation. In this paper, we add CBAM [21] to Mask R-CNN [18] for the first time, which also has channel attention and spatial attention. Channel attention tells us “what” is meaningful and spatial attention tells us “where” important information is.

## **3. Methods**

The Mask R-CNN is extended from Faster R-CNN [18,74] and is a two-stage framework. The first stage is the Region Proposal Network [43] (RPN). In the second stage, further fine-tuning frames for the ROI proposed by RPN. Finally, the parallel Mask head branch will segment the target mask.

In the weakly supervised segmentation branch, we directly use the detection boxes to segment the lesion area, to avoid the gap between the pseudo labels and ground-truth labels. This improves the segmentation performance. The key points and differences of the method are detailed in the following subsections. The architecture of the segmentation model is shown in Figure 2.

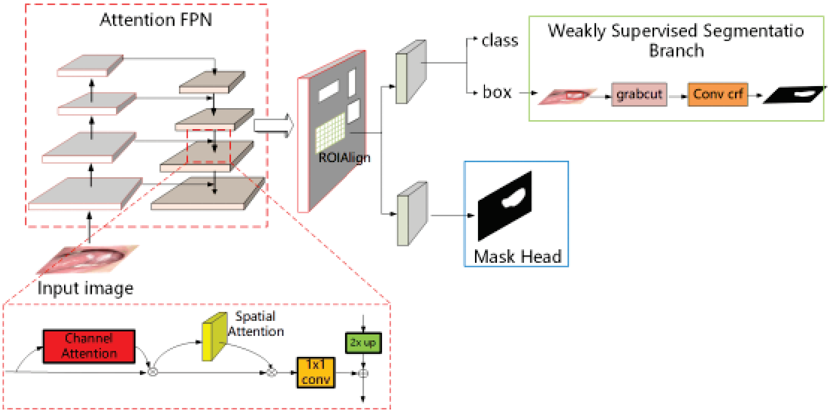


**Figure 2.** Medical lesion segmentation framework based on dual attention mechanism.

### *3.1. Segmentation Model Based on Dual Attention Guidance*

FPN, as the backbone network of Mask R-CNN, performs well in conventional segmentation tasks. However, when segmenting medical images with complex and fuzzy boundaries, it often results in missing segmentation or wrong segmentation. The reason is that the correct features of the lesion area are not extracted. Therefore, we add the dual attention module in the backbone network and the improved attention-FPN structure is shown in the attention FPN part of Figure 2. As shown in Figure 3, the specific joining position of the dual attention module is in the Conv block and identity block of ResNet. Conv block and identity block are the basic modules of ResNet network. Conv block has convolution operation on branches, which can change the number of output channels of the block; Identity block has no operation on the branch, and the number of input and output channels of this block is the same. In our model, a dual attention mechanism is added to these two different blocks.

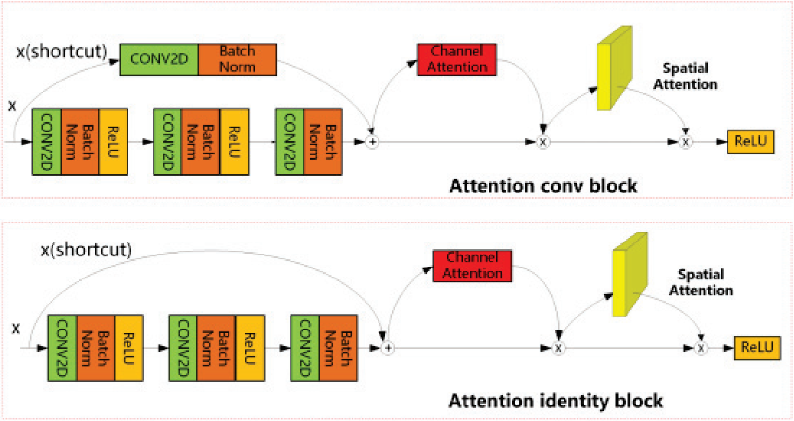


**Figure 3.** Attention conv block and Attention identity block.

In CNN, if the sizes of the convolution kernels are smaller than the step size, the performance of detection and segmentation will decline linearly. FPN is a clever solution by up-sampling high-level features and top-down connecting of low-level features. In the feature extraction and fusion stage, FPN performs well, especially for small target detection. We then use ResNet [75] series in the backbone network to have more flexible choices. Taking ResNets as the backbone network, the FPN network contains three paths, a bottom-up path, a top-down path, and a horizontal connection in the middle. In the forward propagation process of CNN, the feature map after the calculation of the convolution kernel is usually small. The size of the feature map will change after passing through some layers, but not for the other layers. The layers that do not change the size of the feature map are classified as a stage. Specifically, for ResNets, the feature output of the last residual block of each stage is used to activate the output. For conv2, conv3, conv4, and conv5 outputs, the outputs of these final residual blocks are represented as C2, C3, C4, C5, and they have a step size of 4, 8, 16, 32 relatively to the original input image. The top-down process is to perform two times the up-sampling of higher-level features with more abstract and stronger semantic information, and to merge the output results of the up-sampling with the feature map of the previous layer generated from the bottom up through the horizontal connection. After fusion, the high-level features have been strengthened, and the two horizontally connected features should have the same spatial size. This is done to make use of the positioning details of the bottom layer. Each feature map is composed of many channels. In Mask R-CNN, the outputs of ResNet C2, C3, C4, C5 are passed to the next layer of the network for fusion. All channels of these output feature maps are given the same weight, that is, the same attention, but some of these channels are meaningless or erroneous features. We use channel attention after C2, C3, C4, C5 to capture the relationship among global channels. In other words, it encodes different weights for each channel to enhance the weight of important channels and suppress the features of unimportant channels.

In order to calculate the channel attention, the spatial dimensions of the input feature are compressed, the global maximum pooling and global average pooling are performed respectively, and then the multilayer perceptron model (MLP) output features are added and operated through the shared MLP. As shown in Figure 4a, after sigmoid activation operation, the final channel attention map is generated. Multiply the channel attention map and the input feature to generate the input feature of the spatial attention module.

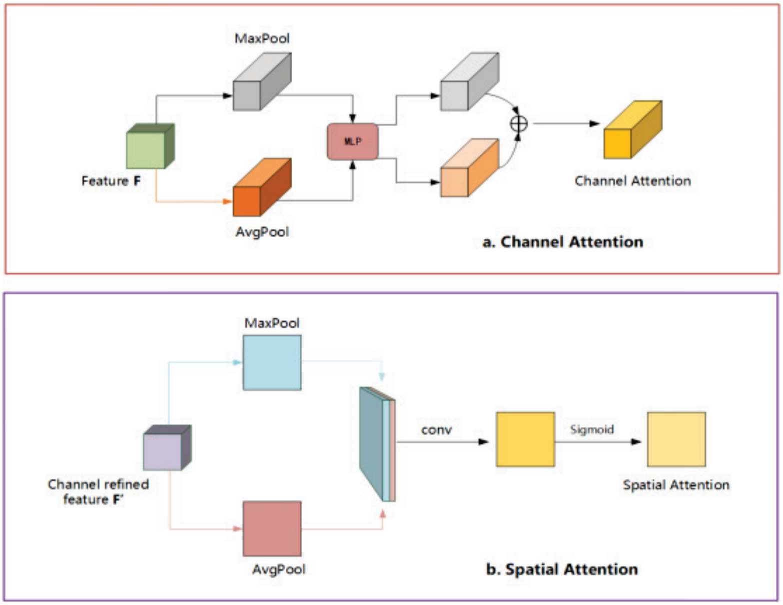


**Figure 4.** The overall structure of dual attention ((**a**). Channel Attention; (**b**). Spatial Attention).

The calculation process is as follows:

$$\begin{aligned}\mathbf{M_c}(\mathbf{F}) &= \sigma(\mathrm{MLP}(\mathrm{AvgPool}(\mathbf{F})) + \mathrm{MLP}(\mathrm{MaxPool}(\mathbf{F}))) \\ &= \sigma\Big(\mathbf{W}_1\Big(\mathbf{W}_0\Big(\mathbf{F}^{\mathrm{c}}_{\mathrm{avg}}\Big)\Big) + \mathbf{W}_1(\mathbf{W}_0(\mathbf{F}^{\mathrm{c}}_{\mathrm{max}}))\Big)\end{aligned} \tag{1}$$

In the above formula, $\mathbf{F^c_{avg}}$ and $\mathbf{F^c_{max}}$ represent the averaged pooling feature and the maximum pooling feature, respectively, and $\sigma$ represents the sigmoid activation function. $M^c \in R^{c\times1\times1}$, $W_0$ and $W_1$ are the weights of MLP.

In the original FPN network, the bottom-up and top-down features are fused directly into the horizontal connection, which lacks the spatial dependence among pixels. We use the spatial relationship among pixels to generate a spatial attention map, making the network pay attention to "where" the information is, which supplements channel attention.

In order to calculate the spatial attention, we use the feature map output by the channel attention module as input to perform global maximum pooling and global average pooling on the channel axis, respectively. We then conduct the concat operation and the $7 \times 7$ convolution operation. Finally, the $1 \times H \times W$ spatial attention map is generated through the sigmoid activation function, as shown in the Figure 4b. The final feature map is obtained by multiplying the spatial attention graph and the input features of this module. Spatial attention is calculated as follows:

$$\begin{aligned}\mathrm{M_s}(\mathrm{F}) &= \sigma\Big(f^{7\times7}([\,\mathrm{AvgPool}(\mathbf{F}); \mathrm{MaxPool}(\mathbf{F})])\Big) \\ &= \sigma\Big(f^{7\times7}\Big(\Big[\mathbf{F}^{\mathrm{s}}_{\mathrm{avg}}; \mathbf{F}^{\mathrm{s}}_{\mathrm{max}}\Big]\Big)\Big).\end{aligned} \tag{2}$$

In the above formula, $\mathbf{F^s_{avg}}$ and $\mathbf{F^s_{max}}$ represent the averaged pooling feature and the maximum pooling feature respectively, and the dimension is $1 \times H \times W$. $\sigma$ represents the sigmoid activation function.

The outputs of ResNet C2, C3, C4, C5 calculate the one-dimensional channel attention $M_c \in R^{c\times1\times1}$ on the channel axis through channel attention module and a two-dimensional spatial attention map $M_s \in R^{1\times H\times W}$ is calculated on the spatial axis through the Spatial Attention module. Then, the final feature map is calculated through the series connection and the process is shown in Figure 3. The training of the fully supervised segmentation network is carried out using a dataset with pixel-level labels. The training process is the same as the original Mask R-CNN. The detection branch and the Mask Head branch will be trained at the same time; the inference phase will generate the final segmentation result in the Mask Head branch.

### *3.2. Weakly Supervised Segmentation*

In this section, we will show the weakly supervised segmentation method. Given a dataset $D = \{I_n, B_n\}_n^N$ with bounding box labels, $N$ represents the number of samples datasets, $I_n$ represents the $n$ th picture, and $B_n$ represents the box-level label of $I_n$. In this section, our goal is to build an end-to-end weakly supervised segmentation model using only box-level dataset D. We know that Mask R-CNN [18] not only has the function of instance segmentation but also has the ability of target detection. In Section 3.1, the improved segmentation model also performs well on target detection, which is an important merit for our weakly supervised segmentation method. Apart from the fully supervised segmentation framework explained before, we abandon the fully supervised mask head branch and add a weakly supervised segmentation branch (Figure 2). The overall segmentation framework is shown in Figure 2. In the inference stage, the detection branch will give the target's tight bounding box, the area outside the bounding box is the background area, and there are some background pixels mixed in the target box. We use GrabCut [55,76,77] to separate the foreground and background in the boundary box. So far, we have obtained preliminary segmentation results, but the foreground segmented by GrabCut [55,78] has holes in the interior, and inaccurate boundaries. In order to obtain better performance, we use the faster ConvCRF [20,61–63] to generate the final segmentation mask.

When using GrabCut to separate the foreground and background according to the detection bounding boxes of the lesion area, the efficiency will be slow with large input image. In order to increase the speed, the image must be zoomed, but the small lesion area will lose a lot of information after the image size reduction, thus, finding a balance between efficiency and effect is necessary. In order to calculate the scale of the zooming, we determine the relative scale of the detection box and the image.

The training process of the weakly supervised segmentation network is different from the fully supervised process. The latter uses the box-level weakly supervised label dataset and only trains and updates the parameters of the detection branch, which is essentially the process of training a target detection network. In the inference stage, the weakly supervised segmentation branch will generate weakly supervised segmentation results based on the output bounding box of the target detection network.

## 4. Result

### *4.1. Experimental Details and Evaluation Strategies*

Experimental details: The proposed method is evaluated on two popular datasets including the OLK dataset and the ISIC [12] 2018. We use the keras framework to implement our model. We use ResNets as our backbone network and fine tune the network from a pre-trained model which is learned on the MS COCO dataset. The batch size, learning rate, weight decay, momentum and Epoch are 2, 0.001, $10^{-4}$, 0.9 and 60, respectively. The optimizer is Adam, and data enhancement, such as rotation, affine transformation, and random clipping, are performed. The framework is trained on a machine with a NVIDIA TITAN RTX 24 GB GPU.

Evaluation strategy: Like most medical image segmentation evaluation strategy, we use the standard F1-score (F1), sensitivity (SEN), specificity (SPE), accuracy (ACC), and Jaccard similarity to evaluate our proposed model.

### *4.2. Oral Leukoplakia Dataset*

Oral leukoplakia is an injury to the oral mucosa and a precancerous lesion. We obtained the oral leukoplakia medical image dataset from the hospital which contains 90 original images and corresponding masks labeled by professional doctors. We divided the whole image dataset into a training set (55 images), a validation set (15 images), and a test set (20 images). Since the number of oral leukoplakia datasets is small, there is no test set. Compared with the ISIC 2018 dataset, the segmentation task of the oral leukoplakia dataset is more challenging. Not only is the number sparse—only 3% of the ISIC 2018 dataset—but also the boundary of the lesion area is more blurred, the shape is irregular and changeable. In the fully supervised segmentation experiment, the ground-truth labels are the binary masks of the original dataset. In the weakly supervised segmentation experiment, the ground-truth labels are the circumscribed rectangles of the binary masks.

Figure 5 shows the segmentation results of our proposed fully supervised and weakly supervised methods on the oral leukoplakia dataset. The results obtained by our fully supervised segmentation method basically remain consistent with the shapes of the ground truths, although the boundaries do not have very good consistency. Att-Deeplab-V3+ can achieve good performance in quantitative evaluation indicators, but it does not have good shape preservation of lesions with the ground truths, and so are the results obtained by Mask RCNN. In addition, our weakly supervised (WS) segmentation method can achieve segmentation results with more overlaps with the lesions in the ground truths. Table 1 shows the quantitative evaluation indicators of all the methods. From the experimental results, we can see that the proposed fully supervised (FS) and weakly supervised (WS) segmentation methods achieve the best performance, and have many improvements over the baseline method (Mask RCNN). In addition, the segmentation performance of the weakly supervised method is very close to that of the fully supervised method. Therefore, the new weakly supervised segmentation model greatly reduces the cost of data annotation for the localization and segmentation of disease regions.

**Table 1.** Performance comparison of the proposed segmentation network and other methods on the Oral leukoplakia dataset.

| | Method | F1 | SEN | SPE | ACC | Jaccard Similarity |
|---|---|---|---|---|---|---|
| FS | Att-Deeplab v3+ [79] | 0.514 | 0.521 | 0.953 | 0.935 | 0.935 |
| | U2-Net [80] | 0.759 | 0.734 | 0.986 | 0.967 | 0.967 |
| | Mask R-CNN [18] | 0.741 | 0.704 | 0.978 | 0.959 | 0.959 |
| | Ours-full | **0.815** | **0.758** | **0.990** | **0.967** | **0.967** |
| WS | Ours-weak | **0.684** | **0.843** | **0.964** | **0.943** | **0.943** |

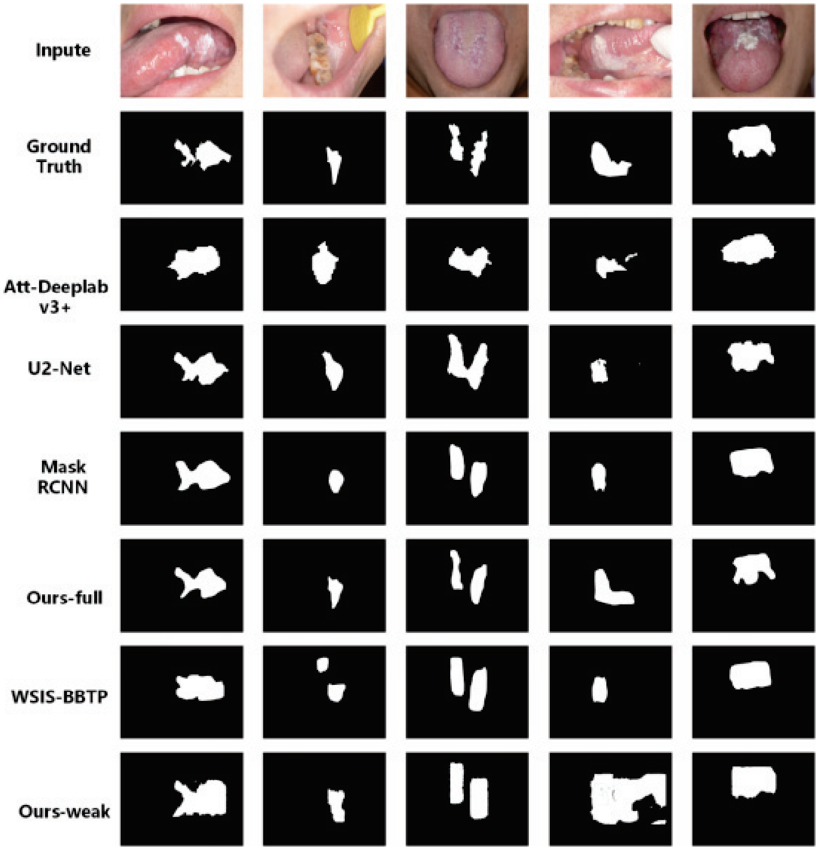


**Figure 5.** Segmentation results of fully supervised and weakly supervised segmentation method on the Oral leukoplakia dataset.

## *4.3. ISIC*

The ISIC 2018 [12] challenge dataset was published by the international skin imaging collaboration (ISIC) in 2018. We select the dermatoscopy image lesion boundary segmentation dataset of challenge task 1, which contains 2594 original images and the corresponding binary ground-truth masks. In order to compare with other methods, we set up the same set with other methods, including 1815 training sets, 259 verification sets and 520 test sets. We set the image input size to 768 × 768. In the fully supervised segmentation experiment, the ground-truth labels are the binary masks of the original dataset. In the weakly supervised segmentation experiment, the ground-truth labels are the circumscribed rectangles of the binary masks. In addition, ISIC 2017 is also a famous skin image dataset similar to ISIC 2018, we also conducted relevant experiments on ISIC 2017 to verify the performance of different models.

Figure 6 shows the segmentation results of the proposed fully supervised and weakly supervised methods on the ISIC 2018 [12] dataset. It is not difficult to find out that some of the methods suffer a serious performance degradation on the oral leukoplakia dataset, but achieved much better performance on the ISIC 2018 dataset. The main reason may be that the lesion regions in the skin disease images are easier segmented than those in oral leukoplakia images. Therefore, there is little difference in the result images by the image segmentation methods. In addition, the weakly supervised segmentation methods also achieved good performance, and their results are almost close to the results obtained by the fully supervised segmentation methods. However, the results of our weakly supervised segmentation method have better shape consistency with the ground truths than WSIS-BBTP. Furthermore, the quantitative evaluation indicators of the experimental results are shown in Table 2. It can be seen that the proposed fully supervised segmentation method achieved great improvements over the original mask R-CNN, and also achieves competitive results compared with other methods. At the same time, our weakly supervised segmentation method has better performance than WSIS-BBTP, and also achieved comparable performance regarding fully supervised segmentation methods, and even surpassed some fully supervised segmentation methods, such as U-net [5], Att U-net [6], R2U-net [81], Att R2U-Net [81], BCDU-Net [82]. Furthermore, the experimental results on ISIC 2017 are shown in Table 3. It can be seen that there have been better performances for different models compared with ISIC 2018, and our weakly supervised segmentation method also achieved a competitive performance with other fully supervised segmentation models.

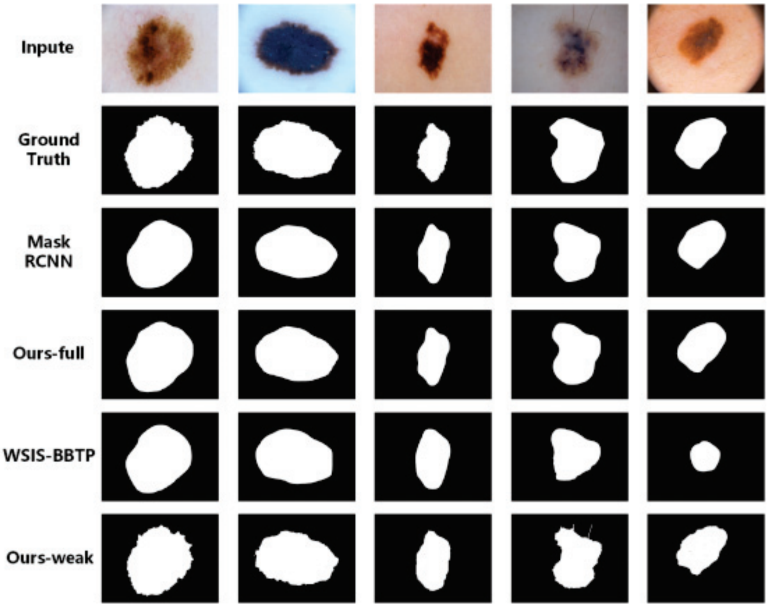


**Figure 6.** Segmentation results of fully supervised and weakly supervised segmentation methods on the ISIC 2018 dataset.

**Table 2.** Performance comparison of the proposed segmentation network and other methods on the ISIC 2018 dataset.

| | Method | F1 | SEN | SPE | ACC | Jaccard Similarity |
|---|---|---|---|---|---|---|
| FS | U-net [5] | 0.647 | 0.708 | 0.964 | 0.890 | 0.549 |
| | Att U-net [6] | 0.665 | 0.717 | 0.967 | 0.897 | 0.566 |
| | R2U-net [81] | 0.679 | 0.792 | 0.928 | 0.880 | 0.581 |
| | Att R2U-Net [81] | 0.691 | 0.726 | 0.971 | 0.904 | 0.592 |
| | BCDU-Net [82] | 0.851 | 0.785 | 0.982 | 0.937 | 0.937 |
| | MCGU-Net [83] | 0.895 | 0.848 | 0.986 | 0.955 | 0.955 |
| | Deeplab v3+ [25] | 0.882 | 0.856 | 0.977 | 0.951 | 0.951 |
| | Att-Deeplab v3+ [79] | 0.712 | 0.875 | **0.988** | **0.964** | **0.964** |
| | Mask R-CNN [18] | 0.872 | 0.846 | 0.974 | 0.947 | 0.947 |
| | Wu's Method [84] | - | **0.942** | 0.941 | 0.947 | - |
| | Ours-full | **0.904** | 0.865 | 0.987 | 0.961 | 0.961 |
| WS | WSIS-BBTP [85] | 0.858 | 0.784 | 0.967 | 0.937 | 0.937 |
| | Ours-weak | **0.874** | **0.861** | **0.986** | **0.950** | **0.950** |

**Table 3.** Performance comparison of the proposed segmentation network and other methods on ISIC 2017 dataset.

| | Method | F1 | SEN | SPE | ACC | Jaccard Similarity |
|---|---|---|---|---|---|---|
| FS | U-net [5] | 0.8682 | **0.9479** | 0.9263 | 0.9314 | 0.9314 |
| | Melanoma det [86] | - | - | - | 0.9340 | - |
| | Lesion Analysis [87] | - | 0.8250 | 0.9750 | 0.9340 | - |
| | R2U-net [81] | 0.8920 | 0.9414 | 0.9425 | 0.9424 | 0.9421 |
| | BCDU-Net [82] | 0.8810 | 0.8647 | 0.9751 | 0.9528 | 0.9528 |
| | MCGU-Net [83] | 0.8950 | 0.8480 | 0.9860 | 0.9550 | 0.9550 |
| | HRFB [88] | **-** | 0.870 | 0.964 | **0.938** | **-** |
| | Deeplab v3+ [25] | 0.9162 | 0.8733 | **0.9921** | 0.9691 | 0.9691 |
| | Att-Deeplab v3+ [79] | **0.9190** | 0.8851 | 0.9901 | **0.9698** | **0.9698** |
| | Mask R-CNN [18] | 0.9092 | 0.8644 | 0.9794 | 0.9472 | 0.9472 |
| | Wu's Method [84] | - | 0.9061 | 0.9628 | 0.9570 | - |
| | Ours-full | 0.9145 | 0.8865 | 0.9879 | 0.9635 | 0.9636 |
| WS | Ours-weak | **0.8845** | **0.8473** | **0.9706** | **0.9384** | **0.9384** |

From these related works, performed for skin lesion segmentation, we can see that our method achieves a competitive performance compared with the classic lesion segmenting methods—Wu's method [84], HRFB [88] and Att-Deeplab V3+ [79]. In addition, our method is different to these related methods. Specifically: (1) The existing methods for skin lesion segmentation are the methods based on the fully supervised learning, while our proposed method can carry out lesion segmentation based on weekly supervised learning using box level annotations; (2) ADAM [84] attention module, which includes Global Average Pooling (GAP) and Pixel Level Correlation (PC), is designed in Wu's method to capture global contextual information. HRFB [88] provides high-resolution feature mapping to preserve spatial details. Att-Deeplab V3+ [79] introduces two levels of attention mechanism based on deeplab V3+ to capture the relationships between a group of features. We introduce the CBAM module into the FPN (Feature Pyramid Networks) network to form an attention FPN, so as to improve the network's perception of multi-scale images.

In summary, Figures 5 and 6 show the results of the comparison of the segmentation details between our method and other methods. It can be seen that after the attention mechanism is involved, the segmentation of lesion area will have fewer false segmentation and missing segmentation, and the segmentation of boundary details will be more accurate than the original network does. These qualitative results are exactly in line with our expectations of joining the dual attention mechanism, allowing the network to pay attention to "what" and "where". However, in the quantitative evaluation, our fully supervised method achieve the second best results regarding Att-Deeplab v3+ in the ISIC dataset, and surpassed other methods in all indicators on the oral leukoplakia dataset. However, in the oral leukoplakia dataset, Att-Deeplab v3+ achieved the worst results. Even our weakly supervised segmentation results surpassed Att-Deeplab v3+. It can be seen that the segmentation framework we proposed can effectively extract the features of the lesion area; thus, it is robust and adaptive to different datasets. In the segmentation task of oral leukoplakia, due to the small amount of data in the oral leukoplakia dataset, the image size is extremely large, and the scale of the lesion area changes greatly, which will lead to the traditional feature extraction network to lose a lot of details after extracting higher-level information. If the lesion area is small, this area will be ignored, leading to missing segmentation. In contrast, our framework backbone network is based on the feature pyramid network of dual attention. The multi-scale network can fuse the high-level features with richer semantic information and the low-level features with higher resolution, and effectively reduce the phenomenon of missing segmentation. In the weakly supervised segmentation method, the detection model with the attention mechanism also greatly improved the ability of locating the lesion area, which provides an accurate bounding box for the segmentation of GrabCut [55].

In addition, we also analyzed the computational complexity of some related segmentation methods, and the results are shown in Table 4. From these results, we can see that our model have similar computational complexity with most segmentation methods.

**Table 4.** The computational complexity of related segmentation methods.

| Method | Params(M) | GFLOPs |
|---|---|---|
| U-Net [5] | 31 | 233 |
| R2U-Net [81] | 75 | 78 |
| Deeplab V3+ [25] | 59 | 67 |
| Attention U-Net [81] | 51 | 55 |
| Wu′ method [84] | 38 | 33 |
| Our model | 44 | 47 |

## 5. Conclusions

In this paper, we propose an end-to-end medical lesion segmentation framework. In this framework, if pixel-level labels are available, we can use the fully supervised branch to obtain more precise segmentation results. If you only have box-level labels, you can still use the weakly supervised branch to obtain better segmentation results. In addition, we add a dual attention mechanism to improve the network segmentation performance. The dual attention mechanism in Mask R-CNN can help the network focus on the features of important regions, but suppress the unimportant features. This mechanism also provides a more accurate bounding box for weakly supervised branches. In addition, the proposed weakly supervised segmentation branch can greatly reduce the gap between labels and pseudo labels, and achieve comparable performance with fully supervised segmentation. Experimental results on the oral dataset and the ISIC 2018 dataset demonstrate the effectiveness of our proposed framework. In this paper, the fully and weakly supervision segmentation branches are used for lesion segmentation separately, rather than integrating into one model. Therefore, we can design an end-to-end weak supervision image segmentation model in the future.

**Author Contributions:** F.X. and P.Z. designed the research and wrote the manuscript; T.J., X.S., P.X., W.Z. and Z.G. contributed to the improvement of our ideas and to the revision of the manuscript; P.Z., J.S., X.S. and G.G. carried out the data collection and research experiments. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Natural Science Foundation of China under grant agreements Nos. 61973250, 61973249, 61876145, 61802335, 61902313, 61802306. Shaanxi Province Science Fund for Distinguished Young Scholars: 2018JC-016. Key Research and Development Program of Shaanxi (Program No.2019GY-012, 2021GY-077, 2021ZDLGY02-06). Cross disciplinary Research Fund of Shanghai Ninth People's Hospital, Shanghai JiaoTong university School of Medicine (JYJC202113). Shaanxi Provincial Department of Education serves local scientific research: 19JC038.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The images in Oral Leukoplakia Dataset are collected from Department of Oral Mucosal Diseases, Shanghai Ninth People's Hospital, we have gotten the patients' consents before taking the images, and these images have no personal information of patients. ISIC 2018 is a public image dataset for medical image segmentation.

**Conflicts of Interest:** All the authors declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Chang, X.; Yu, Y.L.; Yang, Y.; Xing, E.P. Semantic pooling for complex event analysis in untrimmed videos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1617–1632. [CrossRef]
2. Yan, C.; Chang, X.; Li, Z.; Guan, W.; Ge, Z.; Zhu, L.; Zheng, Q. ZeroNAS: Differentiable Generative Adversarial Networks Search for Zero-Shot Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef] [PubMed]
3. Hesamian, M.H.; Jia, W.; He, X.; Kennedy, P. Deep learning techniques for medical image segmentation: Achievements and challenges. *J. Digit. Imaging* **2019**, *32*, 582–596.
4. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
5. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
6. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N.Y.; Kainz, B.; et al. Attention u-net: Learning where to look for the pancreas. *arXiv* **2018**, arXiv:1804.03999.
7. Xiao, X.; Lian, S.; Luo, Z.; Li, S. Weighted res-unet for high-quality retina vessel segmentation. In Proceedings of the 2018 9th International Conference on Information Technology in Medicine and Education (ITME), Hangzhou, China, 19–21 October 2018; pp. 327–331.
8. Wang, C.; Zhao, Z.; Ren, Q.; Xu, Y.; Yu, Y. Dense u-net based on patchbased learning for retinal vessel segmentation. *Entropy* **2019**, *21*, 168. [CrossRef] [PubMed]
9. Ibtehaz, N.; Rahman, M.S. Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation. *Neural Netw.* **2020**, *121*, 74–87. [CrossRef]
10. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. Unet++: Anested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*; Springer: Cham, Switzerland, 2018; pp. 3–11.
11. Wang, Z.; Zou, N.; Shen, D.; Ji, S. Non-local u-nets for biomedical image segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 6315–6322.
12. Codella, N.; Rotemberg, V.; Tschandl, P.; Celebi, M.E.; Dusza, S.; Gutman, D.; Helba, B.; Kalloo, A.; Liopyris, K.; Marchetti, M.; et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). *arXiv* **2019**, arXiv:1902.03368.
13. Nie, L.; Zhang, L.; Meng, L.; Song, X.; Chang, X.; Li, X. Modeling disease progression via multisource multitask learners: A case study with Alzheimer's disease. *IEEE Trans. Neural Netw. Learning Syst.* **2016**, *28*, 1508–1519. [CrossRef] [PubMed]
14. Yuan, D.; Chang, X.; Huang, P.Y.; Liu, Q.; He, Z. Self-supervised deep correlation tracking. *IEEE Trans. Image Process.* **2020**, *30*, 976–985. [CrossRef]
15. Camalan, S.; Mahmood, H.; Binol, H. Convolutional Neural Network-Based Clinical Predictors of Oral Dysplasia: Class Activation Map Analysis of Deep Learning Results. *Cancers* **2021**, *13*, 1291. [CrossRef]
16. Jubair, F.; Al-karadsheh, O.; Malamos, D. A novel lightweight deep convolutional neural network for early detection of oral cancer. *Oral Dis.* **2021**. [CrossRef] [PubMed]
17. Li, Z.; Yao, L.; Chang, X.; Zhan, K.; Sun, J.; Zhang, H. Zero-shot event detection via event-adaptive concept relevance mining. *Pattern Recognit.* **2019**, *88*, 595–603. [CrossRef]
18. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
19. Luo, M.; Chang, X.; Nie, L.; Yang, Y.; Hauptmann, A.G.; Zheng, Q. An adaptive semisupervised feature analysis for video semantic recognition. *IEEE Trans. Cybern.* **2017**, *48*, 648–660. [CrossRef] [PubMed]

20. Ma, Z.; Chang, X.; Yang, Y.; Sebe, N.; Hauptmann, A.G. The many shades of negativity. *IEEE Trans. Multimed.* **2017**, *19*, 1558–1568. [CrossRef]
21. 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.
22. Li, Z.; Nie, F.; Chang, X.; Nie, L.; Zhang, H.; Yang, Y. Rank-constrained spectral clustering with flexible embedding. *IEEE Trans. Neural Netw. Learning Syst.* **2018**, *29*, 6073–6082. [CrossRef] [PubMed]
23. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
24. Pinheiro, P.O.O.; Collobert, R.; Dollar, P. Learning to segment objec candidates. *Adv. Neural Inf. Process. Syst.* **2015**, 28, 1990–1998.
25. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
26. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context encoding for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7151–7160.
27. Ding, H.; Jiang, X.; Shuai, B.; Liu, A.Q.; Wang, G. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2393–2402.
28. Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; et al. Hybrid task cascade for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4974–4983.
29. Xu, D.; Ouyang, W.; Wang, X.; Sebe, N. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 675–684.
30. 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] [PubMed]
31. Ren, P.; Xiao, Y.; Chang, X.; Huang, P.Y.; Li, Z.; Chen, X.; Wang, X. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Comput. Surv.* **2021**, *54*, 1–34. [CrossRef]
32. Lin, G.; Milan, A.; Shen, C.; Reid, I. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1925–1934.
33. Liu, W.; Rabinovich, A.; Berg, A.C. Parsenet: Looking wider to see better. *arXiv* **2015**, arXiv:1506.04579.
34. 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]
35. 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.
36. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 2169–2178.
37. Hayder, Z.; He, X.; Salzmann, M. Boundary-aware instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5696–5704.
38. Dai, J.; He, K.; Sun, J. Instance-aware semantic segmentation via multitask network cascades. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3150–3158.
39. Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. Fully convolutional instance-aware semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2359–2367.
40. 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, 18–23 June 2018; pp. 8759–8768.
41. Chen, L.-C.; Hermans, A.; Papandreou, G.; Schroff, F.; Wang, P.; Adam, H. Masklab: Instance segmentation by refining object detection with semantic and direction features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4013–4022.
42. Yan, C.; Zheng, Q.; Chang, X.; Luo, M.; Yeh, C.H.; Hauptman, A.G. Semantics-preserving graph propagation for zero-shot object detection. *IEEE Trans. Image Process.* **2020**, *29*, 8163–8176. [CrossRef]
43. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Compute: Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
44. 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.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
45. Wang, X.; You, S.; Li, X.; Ma, H. Weakly-supervised semantic segmentation by iteratively mining common object features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1354–1362.

46. Wei, Y.; Xiao, H.; Shi, H.; Jie, Z.; Feng, J.; Huang, T.S. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7268–7277.
47. Huang, Z.; Wang, X.; Wang, J.; Liu, W.; Wang, J. Weakly-supervised semantic segmentation network with deep seeded region growing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7014–7023.
48. Ahn, J.; Kwak, S. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4981–4990.
49. Ge, W.; Yang, S.; Yu, Y. Multi-evidence filtering and fusion for multilabel classification, object detection and semantic segmentation based on weakly supervised learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1277–1286.
50. Wei, Y.; Feng, J.; Liang, X.; Cheng, M.-M.; Zhao, Y.; Yan, S. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1568–1576.
51. Xu, J.; Schwing, A.G.; Urtasun, R. Learning to segment under various forms of weak supervision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3781–3790.
52. Lin, D.; Dai, J.; Jia, J.; He, K.; Sun, J. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3159–3167.
53. Li, Z.; Nie, F.; Chang, X.; Yang, Y. Beyond trace ratio: Weighted harmonic mean of trace ratios for multiclass discriminant analysis. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2100–2110. [CrossRef]
54. Bearman, A.; Russakovsky, O.; Ferrari, V.; Fei-Fei, L. What's the point: Semantic segmentation with point supervision. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 549–565.
55. Rother, C.; Kolmogorov, V.; Blake, A. "grabcut" interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* **2004**, *23*, 309–314. [CrossRef]
56. Pont-Tuset, J.; Arbelaez, P.; Barron, J.T.; Marques, F.; Malik, J. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 128–140. [CrossRef]
57. Verbeek, J.; Triggs, W. Scene Segmentation with CRFs Learned from Partially Labeled Images. In Proceedings of the NIPS 2007—Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007; pp. 1553–1560.
58. He, X.; Zemel, R. Learning hybrid models for image annotation with partially labeled data. *Adv. Neural Inf. Process. Syst.* **2**008, *21*, 625–632.
59. Luo, M.; Chang, X.; Li, Z.; Nie, L.; Hauptmann, A.G.; Zheng, Q. Simple to complex cross-modal learning to rank. *Comput. Vision Image Underst.* **2017**, *163*, 67–77. [CrossRef]
60. Luo, M.; Nie, F.; Chang, X.; Yang, Y.; Hauptmann, A.G.; Zheng, Q. Adaptive unsupervised feature selection with structure regularization. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 944–956. [CrossRef]
61. Teichmann, M.T.; Cipolla, R. Convolutional crfs for semantic segmentation. *arXiv* **2018**, arXiv:1805.04777.
62. Cheng, Z.; Chang, X.; Zhu, L.; Kanjirathinkal, R.C.; Kankanhalli, M. MMALFM: Explainable recommendation by leveraging reviews and images. *ACM Trans. Inf. Syst.* **2019**, *37*, 1–28. [CrossRef]
63. Chang, X.; Nie, F.; Wang, S.; Yang, Y.; Zhou, X.; Zhang, C. Compound rank-*k* projections for bilinear analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1502–1513. [CrossRef]
64. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
65. Gong, C.; Tao, D.; Chang, X.; Yang, J. Ensemble teaching for hybrid label propagation. *IEEE Trans. Cybern.* **2017**, *49*, 388–402. [CrossRef] [PubMed]
66. Zhang, D.; Yao, L.; Chen, K.; Wang, S.; Chang, X.; Liu, Y. Making sense of spatio-temporal preserving representations for EEG-based human intention recognition. *IEEE Trans. Cybern.* **2019**, *50*, 3033–3044. [CrossRef] [PubMed]
67. Zhan, K.; Chang, X.; Guan, J.; Chen, L.; Ma, Z.; Yang, Y. Adaptive structure discovery for multimedia analysis using multiple features. *IEEE Trans. Cybern.* **2018**, *49*, 1826–1834. [CrossRef] [PubMed]
68. Ren, P.; Xiao, Y.; Chang, X.; Huang, P.Y.; Li, Z.; Gupta, B.B.; Wang, X. A Survey of Deep Active Learning. *ACM Comput. Surv.* **2021**, *54*, 1–40. [CrossRef]
69. 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, 18–23 June 2018; pp. 7132–7141.
70. Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; Chua, T.-S. Scacnn: Spatial and channel-wise attention in convolutional networks for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5659–5667.

71. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
72. Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; Liu, W. Ccnet: Crisscross attention for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 603–612.
73. Wang, S.; Chang, X.; Li, X.; Long, G.; Yao, L.; Sheng, Q.Z. Diagnosis code assignment using sparsity-based disease correlation embedding. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 3191–3202. [CrossRef]
74. Chen, K.; Yao, L.; Zhang, D.; Wang, X.; Chang, X.; Nie, F. A semisupervised recurrent convolutional attention model for human activity recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 1747–1756. [CrossRef] [PubMed]
75. 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.
76. Yu, E.; Sun, J.; Li, J.; Chang, X.; Han, X.H.; Hauptmann, A.G. Adaptive semi-supervised feature selection for cross-modal retrieval. *IEEE Trans. Multimed.* **2018**, *21*, 1276–1288. [CrossRef]
77. Ma, Z.; Chang, X.; Xu, Z.; Sebe, N.; Hauptmann, A.G. Joint attributes and event analysis for multimedia event detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 2921–2930. [CrossRef]
78. Li, Z.; Nie, F.; Chang, X.; Yang, Y.; Zhang, C.; Sebe, N. Dynamic affinity graph construction for spectral clustering using multiple features. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 6323–6332. [CrossRef]
79. Azad, R.; Asadi-Aghbolaghi, M.; Fathy, M.; Escalera, S. Attention Deeplabv3+: Multi-level Context Attention Mechanism for Skin Lesion Segmentation. In *Lecture Notes in Computer Science, Proceedings of the Computer Vision—ECCV 2020 Workshops, ECCV 2020, Glasgow, UK, 23–28 August 2020*; Bartoli, A., Fusiello, A., Eds.; Springer: Cham, Switzerland, 2020; doi: 10.1007/978-3-030-66415-2/_16. [CrossRef]
80. Qin, U.; Zhang, Z.; Huang, C.; Dehghan, M.; Osmar, R.Z.; Jagersand, M. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognit.* **2020**, *106*, 107404. [CrossRef]
81. Alom, M.Z.; Hasan, M.; Yakopcic, C.; Taha, T.M.; Asari, V.K. Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation. *arXiv* **2018**, arXiv:1802.06955.
82. Azad, R.; Asadi-Aghbolaghi, M.; Fathy, M.; Escalera, S. Bi-directional ConvLSTM U-Net with Densley connected convolutions. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Seoul, Korea, 27 October–2 November 2019.
83. Asadi-Aghbolaghi, M.; Azad, R.; Fathy, M.; Escalera, S. Multi-level context gating of embedded collective knowledge for medical image segmentation. *arXiv* **2020**, arXiv:2003.05056.
84. Wu, H.; Pan, J.; Li, Z.; Wen, Z.; Qin, J. Automated Skin Lesion Segmentation Via an Adaptive Dual Attention Module. *IEEE Trans. Med. Imaging* **2021**, *40*, 357–370. [CrossRef]
85. Hsu, C.-C.; Hsu, K.-J.; Tsai, C.-C.; Lin, Y.-Y.; Chuang, Y.-Y. Weakly supervised instance segmentation using the bounding box tightness prior. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 6586–6597.
86. Codella, N.C.; Gutman, D.; Celebi, M.E.; Helba, B.; Marchetti, M.A.; Dusza, S.W.; Kalloo, A.; Liopyris, K.; Mishra, N.; Kittler, H.; et al . Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018; pp. 168–172.
87. Li, Y.; Shen, L. Skin lesion analysis towards melanoma detection using deep learning network. *Sensors* **2018**, *18*, 556. [CrossRef] [PubMed]
88. Xie, F.; Yang, J.; Liu, J. Skin lesion segmentation using high-resolution convolutional neural network. *Comput. Methods Programs Biomed.* **2019**, *186*, 105241. [CrossRef] [PubMed]

electronics

MDPI

*Article*

# Multipopulation Particle Swarm Optimization for Evolutionary Multitasking Sparse Unmixing

**Dan Feng [1], Mingyang Zhang [2,*] and Shanfeng Wang [3]**

[1] School of Communications and Information Engineering, Xi’an University of Posts and Telecommunications, Xi’an 710121, China; fengdan@xupt.edu.cn

[2] School of Electronic Engineering, The Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi’an 710071, China

[3] School of Cyber Engineering, Xidian University, Xi’an 710071, China; sfwang@xidian.edu.cn

* Correspondence: myzhang@xidian.edu.cn; Tel.: +86-18602916113

**Abstract:** Recently, the multiobjective evolutionary algorithms (MOEAs) have been designed to cope with the sparse unmixing problem. Due to the excellent performance of MOEAs in solving the NP hard optimization problems, they have also achieved good results for the sparse unmixing problems. However, most of these MOEA-based methods only deal with a single pixel for unmixing and are subjected to low efficiency and are time-consuming. In fact, sparse unmixing can naturally be seen as a multitasking problem when the hyperspectral imagery is clustered into several homogeneous regions, so that evolutionary multitasking can be employed to take advantage of the implicit parallelism from different regions. In this paper, a novel evolutionary multitasking multipopulation particle swarm optimization framework is proposed to solve the hyperspectral sparse unmixing problem. First, we resort to evolutionary multitasking optimization to cluster the hyperspectral image into multiple homogeneous regions, and directly process the entire spectral matrix in multiple regions to avoid dimensional disasters. In addition, we design a novel multipopulation particle swarm optimization method for major evolutionary exploration. Furthermore, an intra-task and inter-task transfer and a local exploration strategy are designed for balancing the exchange of useful information in the multitasking evolutionary process. Experimental results on two benchmark hyperspectral datasets demonstrate the effectiveness of the proposed method compared with the state-of-the-art sparse unmixing algorithms.

**Keywords:** evolutionary multitasking; particle swarm optimization; multipopulation optimization; computational intelligence; sparse unmixing

**Citation:** Feng, D.; Zhang, M.; Wang, S. Multipopulation Particle Swarm Optimization for Evolutionary Multitasking Sparse Unmixing. *Electronics* **2021**, *10*, 3034. https://doi.org/10.3390/electronics10233034

Academic Editor: Amir Mosavi

Received: 22 October 2021
Accepted: 4 December 2021
Published: 5 December 2021

## 1. Introduction

With the progress of remote sensing technology, hyperspectral imagery, which can obtain hundreds of sequential spectrum bands, has been widely applied in both civilian and military scenarios, for example, land-cover classification [1–3], environmental monitoring [4–6] and target detection [7,8], and so forth. However, there remains the problem of mixed pixels due to the low spatial resolution of sensors and the mixture of the surface features [9,10]. Therefore, spectral unmixing aims at extracting the collection of constituent spectra (called endmembers) from the mixed pixels and calculating the fractional abundances of these endmembers [11,12]. Accordingly, different spectral unmixing methods can be divided into three categories, that is, the geometrical-based, statistical-based and sparse-regression-based approaches. Traditional geometrical-based and statistical-based methods are extensively used as they can be utilized easily and flexibly, but they also suffer from the weakness of poor performance on highly mixed scenes spectra and the limitedness of time consumption, respectively [13]. Sparse unmixing, as an emerging spectral unmixing technology in recent years, is devised to find out the optimal solution that can represent each pixel of the hyperspectral image the most from a spectral library known in advance. Among these algorithms, the sparse unmixing via variable splitting and augmented Lagrangian (SUnSAL) based on the alternating direction method of multipliers has been proposed to relax the $l_0$ norm [14]. To overcome the disadvantage of SUnSAL that only utilizes spectral information without considering the spatial-contextual information, Iordache et al. proposed the collaborative SUnSAL (CLSUnSAL) which improves the unmixing results by solving a joint sparse regression problem, where the sparsity is simultaneously imposed to all pixels in the dataset [15,16].

Mathematically, sparse unmixing is an NP-hard problem. Multiobjective evolutionary algorithms (MOEAs), which are able to optimize some contradictory objectives and acquire a set of nondominated solutions called the Pareto-optimal front, are suitable for solving the NP-hard problems and overcoming the aforementioned difficulty in sparse unmixing [17]. A multiobjective sparse unmixing (MOSU) model was first proposed by Gong et al. [18] to deal with the sparse unmixing for hyperspectral imagery. Xu et al. [19] developed a multiobjective optimization based sparse unmixing (SMoSU) to take full advantage of the spectral characteristics of hyperspectral images under the framework of the multiobjective evolutionary algorithm based on decomposition (MOEA/D). In [20], the SMoSU was further improved and a classification-based model called CM-MoSU was designed. The estimation of distribution algorithms is modified to pay more attention to the feasible space with high quality.

However, the existing sparse unmixing algorithms based on MOEAs are limited to the pixel-based unmixing, which leads to the disadvantage of the low efficiency and the lack of the spatial structure information [21]. In some recent studies [22–24], a hyperspectral image is clustered into multiple homogeneous regions based on the assumption that the probability of the active endmember set in the homogeneous region is likely to be the same, which not only reduces the complexity of unmixing, but also further enhances the spatial correlation of pixels in the same category. Interestingly, this coincides with the idea of evolutionary multitasking framework emerging in recent years. The evolutionary multitasking [25] aims to solve different multiobjective optimization problems simultaneously to take advantage of the implicit parallelism from different tasks. Therefore, it is promising to employ the evolutionary multitasking multiobjective framework to efficiently solve the sparse unmixing problem. Besides, the particle swarm optimization (PSO) algorithm, which simulates the regularity of bird cluster activities, has proved to be effective in solving multiobjective endmember extraction problems [26–28]. From this, the current multitasking paradigm can be further explored and applied to sparse unmixing problems.

In this paper, we propose a novel evolutionary multitasking multipopulation particle swarm optimization (EMMPSO) framework for sparse unmixing. In the proposed method, a hyperspectral image is clustered into multiple homogeneous regions first, then the multipopulation particle swarm optimization is employed to explore each sparsity. Finally, the multiobjective optimization is applied to each task simultaneously to obtain a compromise between the reconstruction error and the endmember sparsity. Significantly, it is different from the traditional MOEA-based algorithms that EMMPSO can process the entire matrix due to the decomposition strategy of evolutionary multitasking, aiming at pixel-based unmixing only. In addition, we design a novel intra-task and inter-task transfer strategy to overcome the impact of negative transfer in multitasking. It can not only utilize the effective information in the same task to speed up the convergence of each sub-particle swarm, but also explore the similarities between different tasks to improve the overall convergence performance. Finally, the Pareto optimal solution in each task can be obtained to reverse the final endmember abundance.

The contributions of the proposed EMMPSO algorithm are summarized as follows:

(1) A novel evolutionary multitasking multipopulation particle swarm optimization framework is proposed to solve the sparse unmixing problem. With the decomposition of the evolutionary multitasking, multiple homogeneous regions of a hyperspectral image can be processed simultaneously, which can accelerate the convergence by exploring the relevance of all the tasks. In addition, the Pareto optimal solution

between the reconstruction error and the endmember sparsity can be obtained with the multiobjective optimization.

(2) A multipopulation particle swarm optimization is designed in the multitasking framework for the major evolution. In addition, the intra-task and inter-task transfer strategy are proposed to balance the evolutionary process of exploration and utilization. An efficient local exploration strategy with MOEA is designed to facilitate the search process to obtain the optimal points.

(3) The superiority of EMMPSO on the convergence speed, global optimization performance and unmixing accuracy is substantiated compared with the classical mathemati cal-based and MOEA-basd sparse unmixing algorithms.

The remainder of this paper is structured as follows: Section 2 briefly reviews some related work on sparse unmixing. In Section 3, our method is introduced in detail. Section 4 gives the experimental settings and the analysis of the experimental results. Finally, the conclusions and future works are described in Section 5.

## 2. Related Work

Generally, the mixed pixels are usually unmixed in the linear mixing model. For a single mixed pixel $\boldsymbol{y} \in \mathbb{R}^{L\times 1}$ with $L$ spectral bands, which can be expressed as:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{L\times D}$ is the spectral library. It is worth noting that all the spectral information is known in advance in the spectral library. In addition, $\boldsymbol{x} \in \mathbb{R}^{D\times 1}$ is the corresponding fractional abundance vector, that is, the proportion of each endmember, and $\boldsymbol{n} \in \mathbb{R}^{L\times 1}$ represents the noise term for the mixed pixel. In normal circumstances, a hyperspectral image $\boldsymbol{Y} \in \mathbb{R}^{L\times n}$ contains $n$ pixels, the matrix form of (1) can be formulated as:

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{N}. \tag{2}$$

Therefore, the purpose of sparse unmixing is to obtain the most suitable set of endmembers for the reconstructing remote sensing image from the huge spectral library. Mathematically, this is an NP hard optimization problem, which can be expressed as:

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0, \quad s.t.\ \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 \leq \delta. \tag{3}$$

Many studies employed the relaxation methods to solve the $l_0$-norm problem. SUnSAL [14] resorted to the $l_0$-norm to match $l_0$-norm, and the mathematical optimization formula is as follows:

$$\min_{\boldsymbol{x}} (1/2)\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 + \iota_{R+}(\boldsymbol{x}) + \iota_{\{1\}}(\mathbf{1}^{\mathbf{T}}\boldsymbol{x}), \tag{4}$$

where $\lambda$ stands for a regularization parameter that controls the relative weight between the sparse term and the error term. In [15], the CLSUnSAL takes spatial information into account and directly processes the whole matrix, which is shown as follows:

$$\min_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|_F^2 + \lambda\|\boldsymbol{X}\|_{2,1} + \iota_{R+}(\boldsymbol{X}). \tag{5}$$

Considering the excellent performance of MOEAs in solving NP-hard optimization problems, many studies have turned their attention to MOEAs to solve the sparse unmixing problem in recent year. Gong [18] proposed a novel multiobjective cooperative coevolutionary algorithm to optimize the reconstruction term, the sparsity term and the total variation regularization term simultaneously, which can be expressed as:

$$\min_{\boldsymbol{x}} (\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2, \|\boldsymbol{x}\|_0, \textstyle\sum_{j\in\varepsilon}\|\boldsymbol{x} - \boldsymbol{x}_j\|_1), \tag{6}$$

where $\varepsilon$ stands for the set of the horizontal and vertical neighbors in $\boldsymbol{X}$. Jiang [29] decomposed the sparse unmixing problem into two stages and employed the MOEAs to solve them separately. In the first phase, it is mainly aimed at the endmember extraction, the optimized formula is as follows: $\min\limits_{\boldsymbol{M}}(RSE1, SP1)$, where RSE1 is the residual of the measured hyperspectral image, SP1 represents the size of the measured estimated endmembers ($M$). In the second phase, the extracted abundance estimation becomes the focus, which can be expressed as: $\min\limits_{\boldsymbol{M}}(RSE2, SP2)$, where RSE2 is the residuals of the hyperspectral unmixing, SP2 represents the favorable abundance matrix obtained by incorporating the spatial–contextual information. In addition, Jiang [30] improved the Tp-MoSU to settle the problems of the limited performance in identifying real endmembers from high-noise data in the first phase, and cannot effectively use the spatial context information in the second phase due to the similarity metric used. Besides, many sparse unmixing algorithms based on evolutionary multiobjective decomposition [19,20,31] have also been explored.

Recently, evolutionary multitasking optimization [25,32] has become a new favorite in the field of evolutionary computing. In a nutshell, evolutionary multitasking aims to deal with multiple optimization problems at the same time, and promote the optimization of each task by exploring the hidden relationship between these optimization problems. It is worth noting that many evolutionary multitasking optimization related algorithms have been explored and applied to many fields, such as feature selection [33], reinforcement learning [34] and sparse regression [22] and so forth. In sparse unmixing, a hyperspectral image can be clustered into multiple homogeneous regions according to spatial information, so this coincides with the concept of evolutionary multitasking. It is very promising to model each homogeneous region as an optimization task, though the decomposition of multiple tasks can effectively reduce the impact of dimensional disasters.

## 3. EMMPSO Framework

The pseudo code of EMMPSO is shown in Algorithm 1. In this section, the proposed framework is introduced in detail from initialization, multipopulation particle swarm optimization and the decision making with MOEA.

### 3.1. Initialization and Representation

In sparse unmixing, the spectral library known in advance and a hyperspectral image are input for processing, and the endmember set selected from the library and the corresponding abundance map are output. In the proposed EMMPSO, a hyperspectral image is first clustered into $K$ homogeneous regions, and each homogeneous region is processed as a task [22], which is shown in Figure 1. The spectra of the entire spectrum library are coded into each particle in order, that is, the length of particle is equal to the number of spectra. Considering that the sparsity of particles remains unchanged in the evolution for most current discrete particle swarm optimization algorithms, the population in each task is divided into multiple subpopulations according to the sparsity to ensure that there are particles to explore in each sparsity. For the $s$-th subpopulation in the $j$-th task, the position of each particle is initialized as follows:

$$\{X_{i,s}^{t}\}_j = \{(x_1, x_2, ..., x_n) | x_i \in \{0,1\}, ||X_{i,s}^{t}||_0 = s\}, \tag{7}$$

where $x_i$ is composed of two elements, 0 or 1. If the $x_i$ is equal to 1, it means that the spectrum at the corresponding position in the spectral library is selected, and vice versa.

Then, each particle is evaluated with the reconstruction error ($||\boldsymbol{Y} - \boldsymbol{A}_v\boldsymbol{X}_v||_F$) in the corresponding task, where the $\boldsymbol{Y}$ is the hyperspectral image, $v$ represents the endmember set from the particle $\{X_{i,s}^{t}\}_j$, $\boldsymbol{A}_v$ and $\boldsymbol{X}_v$ are the subset of spectral library $\boldsymbol{A}$ and the abundances of endmembers, respectively. After the evaluation is completed, the skill factor $\tau_{i,s}$, defined as the task with the best performance of the subpopulations with sparsity $s$ in all the tasks, is assigned to each particle. Besides, the $\{pbest_s\}_j$ and $\{gbest_s\}_j$ for the subpopulation with sparsity $s$ in the $j$-th task can be obtained. The velocity of particle is initialized as:

$$\{V_{i,s}^t\}_j = (\{pbest_s\}_j - \{X_{i,s}^t\}_j) + (\{gbest_s\}_j - \{X_{i,s}^t\}_j). \tag{8}$$

**Algorithm 1** The EMMPSO Framework

```
%Initialization
Set $t = 0$, $G = \varnothing$.
for $j = 1$ to $K$ do
    for $s = 1$ to $S$ do
        for $i = 1$ to $N/KS$ do
            $\{X_{i,s}^t\}_j = \{(x_1, x_2, ..., x_n) | x_i \in \{0,1\}, ||X_{i,s}^t||_0 = s\}$.
        end for
    end for
    Evaluate the fitness of each particle in task $T_j$.
    Assign the skill factor $\tau_i$.
    Initialize the $\{pbest_s^t\}_j$ and $\{gbest_s^t\}_j$.
    $\{V_{i,s}^t\}_j = (\{pbest_s^t\}_j - \{X_{i,s}^t\}_j) + (\{gbest_s^t\}_j - \{X_{i,s}^t\}_j)$.
    $G_j^t = \sum_{s=1}^{S} \{gbest_s^t\}_j$.
end for
%Evolution
while $t < Maxt$ do
    for $j = 1$ to $K$ do
        Update the $\{X_{i,s}^{t+1}\}_j$ and $\{V_{i,s}^{t+1}\}_j$ based on (9).
    end for
    Update the particle according to Algorithm 2.
    Evaluate the fitness of each particle.
    Update the $\{pbest_s^t\}_j$, $\{gbest_s^t\}_j$, and $G_j^t$ with the Local Exploration Strategy.
    $t = t + 1$.
end while
%Decision Making
Obtain the optimal point in each task from $G_j$.
```

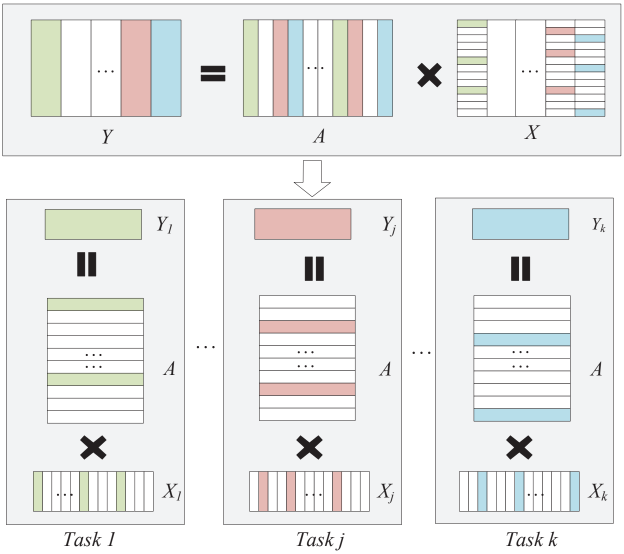


**Figure 1.** The evolutionary multitasking optimization framework for hyperspectral sparse unmixing.

### *3.2. Multipopulation Particle Swarm Optimization for Knowledge Transfer*

Considering the discreteness of decision variables in sparse unmixing, the population in each task is divided into multiple subpopulations according to the sparsity during initialization. In the process of particle swarm evolution, the position and velocity of the particles in the $j$-th task with the sparsity $s$ are updated as follows:

$$\begin{aligned} \{X_{i,s}^{t+1}\}_j &= \{X_{i,s}^{t}\}_j + \{V_{i,s}^{t}\}_j, \\ \{V_{i,s}^{t+1}\}_j &= \begin{cases} T(\{V_{i,s}^{t+1}\}_j), \text{ if } (\text{any}(\{V_{i,s}^{t}\}_j) \geq 0) \\ R(\{X_{i,s}^{t+1}\}_j), \text{ otherwise}, \end{cases} \end{aligned} \tag{9}$$

where $T$ and $R$ are both the selection functions [35].

After updating the positions and velocities of all the particles, we designed an efficient knowledge transfer of intra-task and inter-task to explore the useful information, which is shown in Algorithm 2. Firstly, two particles are randomly selected from the current generation of particles. In the intra-task transfer, the same positions of the particles focus on exploitation. $\cap(p_a, p_b)$ represents the positions where the elements in $p_a$ and $p_b$ are both 1. Then, the new particles directly inherit positions in $\cap(p_a, p_b)$, and the remaining randomly inherit the position on the original particle. On the contrary, the exploration of randomness focuses on the inter-task transfer. $\cup(p_a, p_b)$ represents the positions where the elements are equal to 1 in $p_a$ or $p_b$. For the new particles $p_{a'}$ and $p_{b'}$, $||p_a||_0$ and $||p_b||_0$ positions are directly selected from $\cup(p_a, p_b)$, respectively. Then the $p_a$ and $p_b$ are updated with the better fitness particles. In order to more intuitively illustrate the essence of Algorithm 2, Figure 2 shows a simple example for the genetic knowledge transfer. Two particles with sparsity of 3 and 4 are selected from the current generation first, in the intra-task transfer, the new particles are updated by inheriting all positions in their same positions which refer to the positions in $p_c$, then randomly set the rest of positions to 1 to ensure that the sparsity of the new particles is the same as the previous particles. Similar operations are also performed in the inter-task transfer, but the difference is that the new particles are updated by selecting form the positions with 1 in the previous particle, and randomly set them to 1 with the same sparsity.

**Algorithm 2** Genetic Knowledge Transfer

```
Input: $P^t$: the current generation of particles.
for $g = 1$ to $N/2$ do
    Randomly select two particles $p_a$ and $p_b$ in $P^t$.
    if $\tau_a = \tau_b$ then
        %Intra-task Transfer
        $p_c \leftarrow \cap(p_a, p_b)$.
        $p_{a'} \leftarrow$ Inherit all positions of $p_c$ and randomly set $(||p_a||_0 - ||p_c||_0)$ positions to 1.
        $p_{b'} \leftarrow$ Inherit all positions of $p_c$ and randomly set $(||p_b||_0 - ||p_c||_0)$ positions to 1.
    else
        %Inter-task Transfer
        $p_c \leftarrow \cup(p_a, p_b)$.
        $p_{a'} \leftarrow$ Randomly select $||p_a||_0$ positions in $p_c$.
        $p_{b'} \leftarrow$ Randomly select $||p_b||_0$ positions in $p_c$.
    end if
    Evaluate the fitness of $p_{a'}$ and $p_{b'}$.
    Update the $p_a$ and $p_b$.
    $g = g + 1$.
end for
```

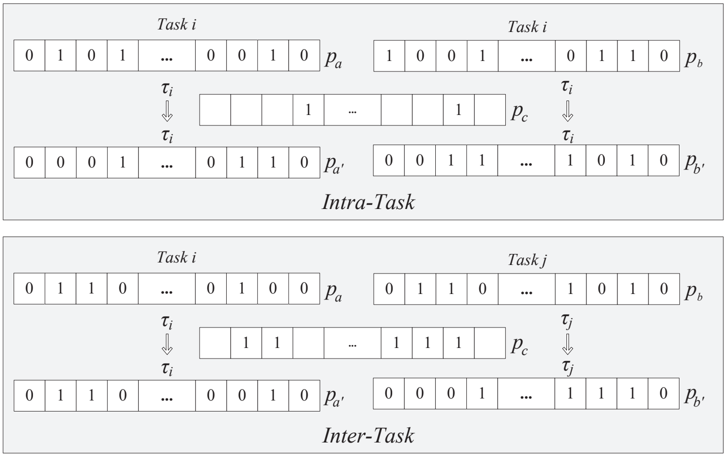

Task i
0 1 0 1 ... 0 0 1 0 $p_a$
Task i
1 0 0 1 ... 0 1 1 0 $p_b$
$\tau_i$
$\tau_i$
1 ... 1 $p_c$
$\tau_i$
$\tau_i$
0 0 0 1 ... 0 1 1 0 $p_{a'}$
0 0 1 1 ... 1 0 1 0 $p_{b'}$
Intra-Task
Task i
0 1 1 0 ... 0 1 0 0 $p_a$
Task j
0 1 1 0 ... 1 0 1 0 $p_b$
$\tau_i$
$\tau_i$
1 1 ... 1 1 1 $p_c$
$\tau_j$
$\tau_j$
0 1 1 0 ... 0 0 1 0 $p_{a'}$
0 0 0 1 ... 1 1 1 0 $p_{b'}$
Inter-Task

**Figure 2.** An example of the knowledge transfer.

### 3.3. An Efficient Local Exploration Strategy with MOEA

After the optimization of multipopulation particle swarms, the set of globally optimal particles with all the sparsity levels on each task ($G = \{\sum_{j=1}^{K}\{\sum_{s=1}^{S}\{gbest_s\}_j\}$) can be obtained. Two conflicting parameters are included in each particle, that is, the endmember sparsity and the reconstruction error. Therefore, we employ the multiobjective optimization algorithm to facilitate the search process to obtain the optimal points in each task. In the evolutionary multitasking multiobjective framework, the optimized function is expressed as follows:

$$\begin{cases} \{\boldsymbol{X}_1^*, \boldsymbol{X}_2^*, \ldots, \boldsymbol{X}_K^*\} = \arg\min\{F(\boldsymbol{X}_1), F(\boldsymbol{X}_2), \ldots, F(\boldsymbol{X}_K)\}, \\ F(\boldsymbol{X}_j) = \min\limits_{\boldsymbol{X}_j}(||\boldsymbol{X}_j||_0, ||\boldsymbol{Y}_j - \boldsymbol{A}\boldsymbol{X}_j||_F), \end{cases} \tag{10}$$

where the $\boldsymbol{Y}_j$ and $\boldsymbol{X}_j$ represent the original image and inversion abundance in the $j$-th task, respectively.

The local exploration strategy processes are in Figure 3. First, the globally optimal particles are transcoded to the first generation of the evolutionary algorithm for the NSGA-II framework. The roulette selection, single-point crossover and bitwise mutation operators are employed to participate in the evolution of multiobjective optimization. Then, the generated offspring are evaluated to update the Pareto front in each task according to the nondominated sorting and crowding distance, and the nondominated solutions are transcoded back to the globally optimal particles.

With the above design, the optimal point in each task can finally be obtained by: $\boldsymbol{X}_{jv}^* = \arg\min ||\boldsymbol{Y}_j - \boldsymbol{A}_v \boldsymbol{X}_{jv}||_F$, which can be solved simply with the least squares method. Finally, the optimal abundance map obtained from each task constitutes the final inverted abundance map.

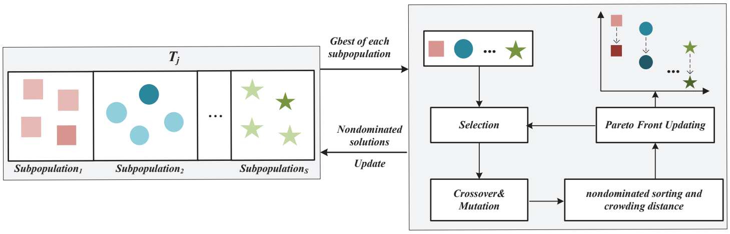
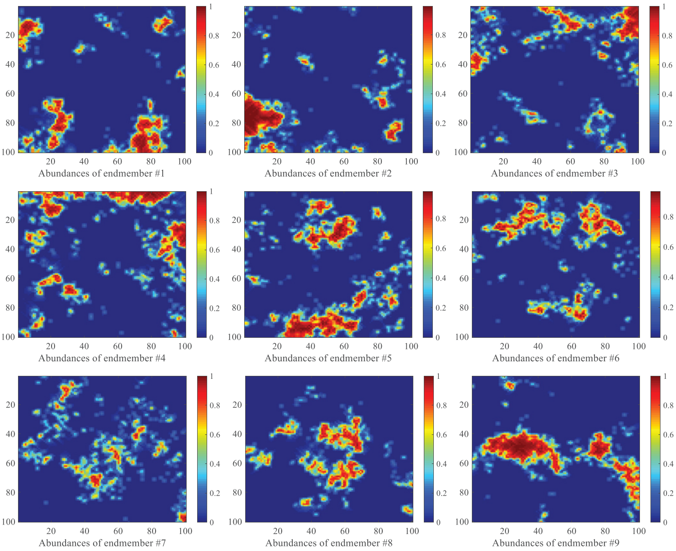


**Figure 3.** The illustration of the Local Exploration Strategy with MOEA.

## 4. Experimental Results

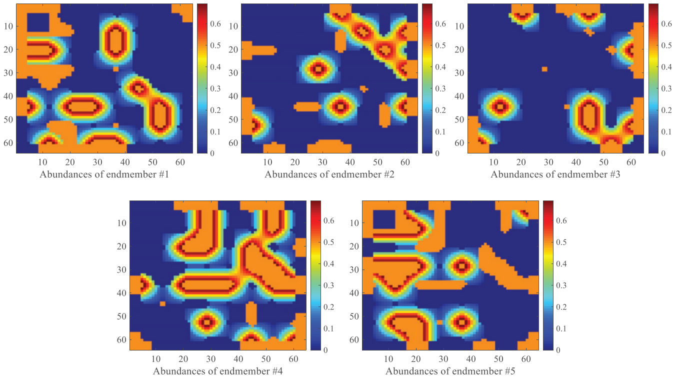
### *4.1. Data Sets*

Data 1 provided by Iordache et al. [36] is an image which contains 100 × 100 pixels and 224 bands in each pixel, and the related abundance map of nine endmembers is shown Figure 4. It contains nine randomly selected signatures from a sublibrary of 230 spectral signals, and the fractional abundances are piecewise smooth. Data 2 provided by Tang et al. [37] is an image which contains 64 × 64 pixels and 224 bands in each pixel, the related abundance map of five endmembers is shown Figure 5. It includes five endmembers from a sublibrary of 498 spectral signals, and the fractional abundances are also homogeneous. These two benchmark datasets were tested at different levels of white noise, that is, SNR = 20, 30 and 40 dB. The number of tasks was set to three on these two datasets as recommended in [22]. In order to maintain the fairness of the experiments, all experimental results were taken from the average results of 20 experiments, which is the same as in the comparative method paper.



Abundances of endmember #1
Abundances of endmember #2
Abundances of endmember #3
Abundances of endmember #4
Abundances of endmember #5
Abundances of endmember #6
Abundances of endmember #7
Abundances of endmember #8
Abundances of endmember #9

**Figure 4.** True abundance maps of five endmembers in data 1.



Abundances of endmember #1
Abundances of endmember #2
Abundances of endmember #3
Abundances of endmember #4
Abundances of endmember #5

**Figure 5.** True abundance maps of nice endmembers in data 2.

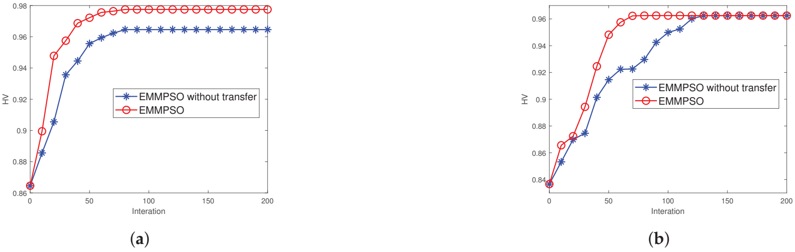
### *4.2. Performance Analysis of EMMPSO*

In this section, the ablation experiments were performed to demonstrate the effectiveness of the knowledge transfer and the local exploration strategy. The hypervolume indicator was used to compare the evolution process and the convergence procedure of the EMMPSO and the EMMPSO without transfer. Hypervolume was calculated using a reference point 1% larger in every component than the corresponding nadir point [38]. As an important indicator to measure the Pareto-optimal front (PF), the larger the value of the hypervolume, or the faster the convergence speed of the hypervolume, the better the PF obtained by the algorithms. The evolution of the hypervolume indicator is shown in Figure 6. It is clear that, after a few iterations, our method can obtain the higher hypervolume values with the help of the intra-task and inter-task transfer strategy. When several related tasks are optimized simultaneously under the framework of evolutionary multitasking, the convergence rate is improved significantly.

Secondly, to test the efficiency of the local exploration, the performance of EMMPSO and EMMPSO without local exploration was compared. Usually, signal to reconstruction error (SRE) is used to measure the quality of the reconstruction of a signal. Table 1 shows the SRE (dB) with different noise levels of our proposed method and the EMMPSO without the local exploration on the simulated data. It can be observed that our method can achieve values of SRE (dB) higher than the EMMPSO without local exploration on both simulated datas. It is obvious to see that the local exploration is useful for facilitating the search process to obtain the optimal points.

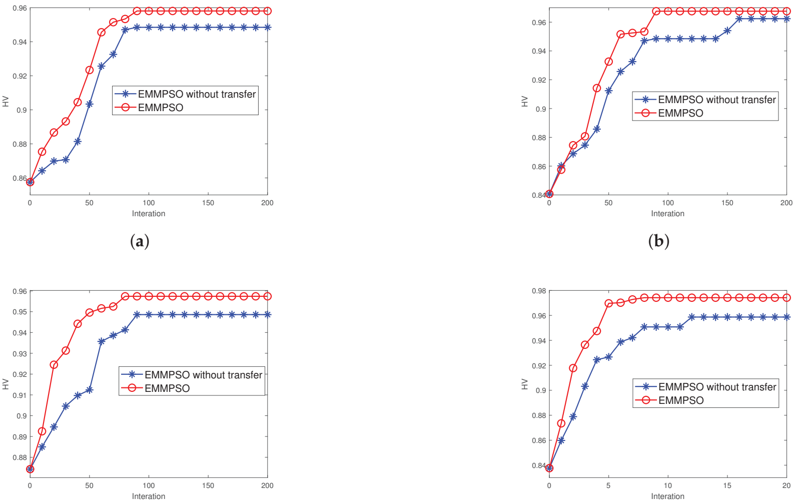
**Table 1.** Comparison of EMMPSO and EMMPSO without Local Exploration on data 1 and data 2.

| **Data 1** | **SRE (dB)** | | |
|---|---|---|---|
| | **20** | **30** | **40** |
| EMMPSO without LE | 7.9435 | 13.3654 | 22.7536 |
| EMMPSO | **8.2783** | **15.8039** | **25.2174** |
| **Data 2** | **SRE (dB)** | | |
| | **20** | **30** | **40** |
| EMMPSO without LE | 10.7025 | 14.7089 | 17.0224 |
| EMMPSO | **12.3572** | **20.5891** | **25.7204** |



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

**Figure 6.** *Cont.*



(**a**)

(**b**)

(**c**)

(**d**)

**Figure 6.** Comparison of the hypervolume indicator for EMMPSO and EMMPSO without transfer. (**a**) task 1 on data 1, (**b**) task 2 on data 1, (**c**) task 3 on data 1, (**d**) task 1 on data 2, (**e**) task 2 on data 2, (**f**) task 3 on data 2.

### *4.3. Comparing with State-of-Art Algorithms*

In order to reflect the superiority of our proposed algorithm, EMMPSO compares with the state-of-art algorithms, including SUnSAL, CLSUnSAL, two-phase multiobjective sparse unmixing (Tp-MOSU) and evolutionary multitasking sparse reconstruction (MTSR). Among them, SUnSAL and CLSUnSAL are the traditional pixel-based and matrix-based processing algorithms. Tp-MOSU and MTSR are two algorithms based on the multiobjective optimization and multitasking optimization, respectively. In order to reflect the advantage of the proposed method, Figures 7 and 8 depict the estimated abundance maps for the endmember 2, 5, 8 on data 1 and the endmember 1, 3, 5 data 2, respectively. The rightmost column represents the abundance map of the real endemembers. The closer the inverted abundance map is to the real abundance map, the better the unmixing performance of the modified algorithm is. It can be seen that the Tp-MOSU, MTSR and EMMPSO exhibit better performances than the other two methods in the similarity with the original abundance map. Although the abundance maps obtained by the Tp-MOSU, MTSR and EMMPSO are similar, the abundance map of EMMPSO has much less noise. Table 2 shows the results of SRE (dB) obtained by the five methods on data 1 and data 2. At different levels of noise, the proposed EMMPSO can always achieve the highest values of SRE (dB) on both simulated datasets. The experimental results on two datasets have proved that our proposed EMMPSO is able to achieve a competitive performance by evolutionary multitasking and local exploration strategy.

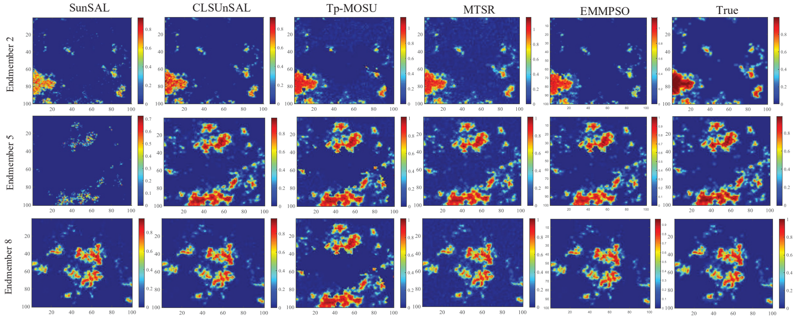


**Figure 7.** The fractional abundance maps of endmember 2, 5, 8 by SunSAL, CLSUnSAL, Tp-MOSU, MTSR and EMMPSO on Data 1.

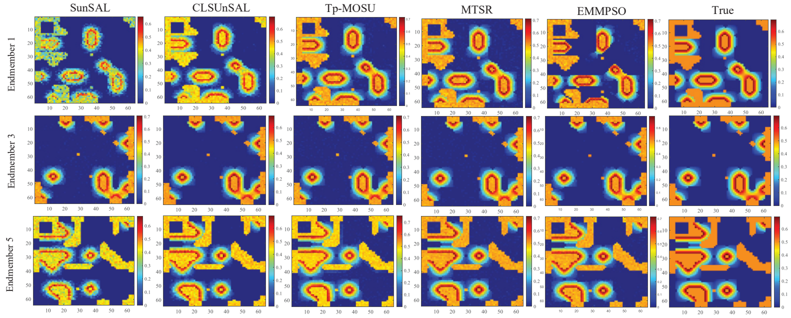


**Figure 8.** The fractional abundance maps of endmember 1, 3, 5 by SunSAL, CLSUnSAL, Tp-MOSU, MTSR and EMMPSO on Data 2.

**Table 2.** Comparison of EMMPSO and other methods on data 1 and data 2.

| **Data 1** | **SRE (dB)** | | |
|---|---|---|---|
| | **20** | **30** | **40** |
| SunSAL | 4.5568 | 8.5833 | 12.9890 |
| CLSUnSAL | 5.5164 | 11.4842 | 18.7935 |
| Tp-MOSU | 8.4083 | 14.4070 | 22.5478 |
| MTSR | 7.0496 | 13.7802 | 22.7329 |
| EMMPSO | **8.5783** | **15.2039** | **24.2174** |
| **Data 2** | **SRE (dB)** | | |
| | **20** | **30** | **40** |
| SunSAL | 3.5823 | 8.0323 | 12.9896 |
| CLSUnSAL | 8.2382 | 13.0988 | 14.3502 |
| Tp-MOSU | 11.3578 | 15.7132 | 18.0457 |
| MTSR | 10.7254 | 14.6143 | 17.6775 |
| EMMPSO | **12.3572** | **20.5891** | **25.7204** |

## 5. Conclusions

In this paper, we propose a novel evolutionary multitasking multiobjective particle swarm optimization framework called EMMPSO to solve the sparse unmixing problem. With processing multiple homogeneous regions of a hyperspectral image simultaneously, the evolution convergence is accelerated. The local exploration strategy with MOEA is also designed to obtain the optimal solution. For the case study, the proposed EMMPSO is compared with some state-of-the-art methods on benchmark simulated datasets. The results demonstrate the superiority of the EMMPSO.

In future work, we will focus on reducing the time complexity of EMMPSO, and design an efficient multiobjective particle swarm optimization paradigm for the sparse unmixing problem.

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

**Funding:** This research received no external funding.

**Acknowledgments:** This work was supported by the National Natural Science Foundation of China under Grant 61906147 and Grant 61806153, the Fundamental Research Funds for the Central Universities (Grant no. XJS200216) and China Post-Doctoral Science Foundation (Grant no. 2021T140528).

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

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| SUnSAL | sparse unmixing algorithm via variable splitting and augmented Lagrangian |
| CLSUnSAL | collaborative SUnSAL |
| MOEAs | Multiobjective evolutionary algorithms |
| MOSU | multiobjective sparse unmixing |
| MOEA/D | multiobjective evolutionary algorithm based on decomposition |
| PSO | particle swarm optimization |
| EMMPSO | evolutionary multitasking multipopulation particle swarm optimization |
| Tp-MOSU | two-phase multiobjective sparse unmixing |
| MTSR | multitasking sparse reconstruction |
| SRE | signal to reconstruction error |

## References

1. Lv, Z.Y.; Liu, T.F.; Zhang, P.; Benediktsson, J.A.; Lei, T.; Zhang, X. Novel Adaptive Histogram Trend Similarity Approach for Land Cover Change Detection by Using Bitemporal Very-High-Resolution Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9554–9574. [CrossRef]
2. Maniatis, D.; Dionisio, D.; Guarnieri, L.; Marchi, G.; Mollicone, D.; Morales, C.; Sanchez-Paus Díaz, A. Toward a More Representative Monitoring of Land-Use and Land-Cover Dynamics: The Use of a Sample-Based Assessment through Augmented Visual Interpretation Using Open Foris Collect Earth. *Remote Sens.* **2021**, *13*, 4197. [CrossRef]
3. Wu, Y; Ma, W.P.; Gong, M.G.; Su, L.Z.; Jiao, L.C. A novel point-matching algorithm based on fast sample consensus for image registration. *IEEE Geosci. Remote Sens. Lett.* **2014**, *12*, 43–47. [CrossRef]
4. Di Biase, V.; Hanssen, R.F. Environmental Strain on Beach Environments Retrieved and Monitored by Spaceborne Synthetic Aperture Radar. *Remote Sens.* **2021**, *13*, 4208. [CrossRef]
5. Jackisch, R.; Lorenz, S.; Zimmermann, R.; Möckel, R.; Gloaguen, R. Drone-Borne Hyperspectral Monitoring of Acid Mine Drainage: An Example from the Sokolov Lignite District. *Remote Sens.* **2018**, *10*, 385. [CrossRef]
6. Wu, Y.; Xiao, Z.; Liu, S.; Miao, Q.; Ma, W.; Gong, M.; Xie, F.; Zhang, Y. A Two-Step Method for Remote Sensing Images Registration Based on Local and Global Constraints. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 5194–5206. [CrossRef]
7. Guo, Y.; Du, L.; Lyu, G. SAR Target Detection Based on Domain Adaptive Faster R-CNN with Small Training Data Size. *Remote Sens.* **2021**, *13*, 4202. [CrossRef]
8. Li, H.; Li, J.; Zhao, Y.; Gong, M.; Zhang, Y.; Liu, T. Cost-Sensitive Self-Paced Learning with Adaptive Regularization for Classification of Image Time Series. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 11713–11727. [CrossRef]
9. Zhang, J.; Zhang, X.; Jiao, L. Sparse Nonnegative Matrix Factorization for Hyperspectral Unmixing Based on Endmember Independence and Spatial Weighted Abundance. *Remote Sens.* **2021**, *13*, 2348. [CrossRef]
10. Lv, Z.; Liu, T.; Wan, Y.; Benediktsson, J.A.; Zhang, X. Post-Processing Approach for Refining Raw Land Cover Change Detection of Very High-Resolution Remote Sensing Images. *Remote Sens.* **2018**, *10*, 472. [CrossRef]
11. Feng, R.; Wang, L.; Zhong, Y. Joint Local Block Grouping with Noise-Adjusted Principal Component Analysis for Hyperspectral Remote-Sensing Imagery Sparse Unmixing. *Remote Sens.* **2019**, *11*, 1223. [CrossRef]
12. Wang, Z.; Wei, J.; Li, J.; Li, P.; Xie, F. Evolutionary Multiobjective Optimization with Endmember Priori Strategy for Large-Scale Hyperspectral Sparse Unmixing. *Electronics* **2021**, *10*, 2079. [CrossRef]
13. Qi, L.; Li, J.; Wang, Y.; Gao, X. Region-Based Multiview Sparse Hyperspectral Unmixing Incorporating Spectral Library. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1140–1144. [CrossRef]

14. Bioucas-Dias, J.M.; Figueiredo, M.A. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In Proceedings of the 2nd Workshop Hyperspectral Image Signal Process, Reykjavik, Iceland, 14–16 June 2010, pp. 1–4.
15. Iordache, M.D.; Bioucas-Dias, J.M.; Plaza, A. Collaborative sparse regression for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 341–354. [CrossRef]
16. Wu, Y.; Li, J.H.; Yuan, Y.Z.; Qin, AK.; 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]
17. Orosz, T.; Rassõlkin, A.; Kallaste, A.; Arsénio, P.; Pánek, D.; Kaska, J.; Karban, P. Robust design optimization and emerging technologies for electrical machines: Challenges and open problems. *Appl. Sci.* **2020**, *10*, 6653. [CrossRef]
18. Gong, M.; Li, H.; Luo, E.; Liu, J.; Liu, J. A multiobjective cooperative coevolutionary algorithm for hyperspectral sparse unmixing. *IEEE Trans. Evol. Comput.* **2016**, *21*, 234–248. [CrossRef]
19. Xu, X.; Shi, Z.; Pan, B. $l_0$-based sparse hyperspectral unmixing using spectral information and a multi-objectives formulation. *ISPRS J. Photogramm. Remote Sens.* **2018**, *141*, 46–58. [CrossRef]
20. Xu, X.; Shi, Z.; Pan, B.; Li, X. A classification-based model for multi-objective hyperspectral sparse unmixing. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9612–9625. [CrossRef]
21. Li, J.; Li, H.; Liu, Y.; Gong, M. Multi-fidelity evolutionary multitasking optimization for hyperspectral endmember extraction. *Appl. Soft Comput.* **2021**, *111*, 107713. [CrossRef]
22. Li, H.; Ong, Y.S.; Gong, M.; Wang, Z. Evolutionary Multitasking Sparse Reconstruction: Framework and Case Study. *IEEE Trans. Evol. Comput.* **2019**, *23*, 733–747. [CrossRef]
23. Li, J.; Du, Q.; Li, Y. Region-based collaborative sparse unmixing of hyperspectral imagery. *Proc. Remotely Sens. Data Compression Commun. Process.* **2016**, *9874*, 127–132.
24. Martin, G.; Plaza, A. Region-Based Spatial Preprocessing for Endmember Extraction and Spectral Unmixing. *IEEE Geosci. Remote Sens. Lett.* **2011**, *8*, 745–749. [CrossRef]
25. Gupta, A.; Ong, Y.S.; Feng, L.; Tan, K.C. Multiobjective Multifactorial Optimization in Evolutionary Multitasking. *IEEE Trans. Evol. Comput.* **2017**, *47*, 1652–1665. [CrossRef]
26. Zhang, B.; Sun, X.; Gao, L.; Yang, L. Endmember Extraction of Hyperspectral Remote Sensing Images Based on the Discrete Particle Swarm Optimization Algorithm. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 4173–4176. [CrossRef]
27. Liu, R.; Zhang, L.; Du, B. A Novel Endmember Extraction Method for Hyperspectral Imagery Based on Quantum-Behaved Particle Swarm Optimization. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2017**, *10*, 1610–1631. [CrossRef]
28. Xu, M.; Zhang, L.; Du, B.; Zhang, L.; Fan, Y.; Song, D. A Mutation Operator Accelerated Quantum-Behaved Particle Swarm Optimization Algorithm for Hyperspectral Endmember Extraction. *Remote Sens.* **2017**, *9*, 197. [CrossRef]
29. Jiang, X.; Gong, M.; Li, H.; Zhang, M.; Li, J. A two-phase multiobjective sparse unmixing approach for hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 508–523. [CrossRef]
30. Jiang, X.; Gong, M.; Zhan, T.; Sheng, K.; Xu, M. Efficient Two-Phase Multiobjective Sparse Unmixing Approach for Hyperspectral Data. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2021**, *14*, 2418–2431. [CrossRef]
31. Pan, B.; Shi, Z.; Xu, X. Multiobjective-Based Sparse Representation Classifier for Hyperspectral Imagery Using Limited Samples. *IEEE Trans. Geosci. Remote. Sens.* **2019**, *57*, 239–249. [CrossRef]
32. Tuysuzoglu, G.; Birant, D.; Pala, A. Majority voting based multi-task clustering of air quality monitoring network in Turkey. *Appl. Sci.* **2019**, *9*, 1610. [CrossRef]
33. Zhang, N.; Gupta, A.; Chen, Z.; Ong, Y.S. Evolutionary Machine Learning with Minions: A Case Study in Feature Selection. *IEEE Trans. Evol. Comput.* **2021**. [CrossRef]
34. Martinez, A.D.; Del Ser, J.; Osaba, E.; Herrera, F. Adaptive Multi-factorial Evolutionary Optimization for Multi-task Reinforcement Learning. *IEEE Trans. Evol. Comput.* **2021**. [CrossRef]
35. Tong, L.; Du, B.; Liu, R.; Zhang, L. An Improved Multiobjective Discrete Particle Swarm Optimization for Hyperspectral Endmember Extraction. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 7872–7882. [CrossRef]
36. Iordache, M.D.; Bioucas-Dias, J.M.; Plaza, A. Total variation spatial regularization for sparse hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 4484–4502. [CrossRef]
37. Tang, W.; Shi, Z.; Wu, Y.; Zhang, C. Sparse unmixing of hyperspectral data using spectral a priori information. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 770–783. [CrossRef]
38. Seada, H.; Deb, K. A unified evolutionary optimization procedure for single, multiple, and many objectives. *IEEE Trans. Evol. Comput.* **2015**, *20*, 358–369. [CrossRef]

*electronics*

MDPI

*Article*

# Reliable Memory Model for Visual Tracking

**Daohui Ge [1], Ruyi Liu [1], Yunan Li [1] and Qiguang Miao [1,2,*]**

[1] School of Computer Science and Technology, Xidian University, Xi'an 710071, China; dhge@stu.xidian.edu.cn (D.G.); ruyiliu@xidian.edu.cn (R.L.); yunanli@xidian.edu.cn (Y.L.)
[2] Xi'an Key Laboratory of Big Data and Intelligent Vision, Xidian University, Xi'an 710071, China
* Correspondence: qgmiao@mail.xidian.edu.cn

**Abstract:** Effectively learning the appearance change of a target is the key point of an online tracker. When occlusion and misalignment occur, the tracking results usually contain a great amount of background information, which heavily affects the ability of a tracker to distinguish between targets and backgrounds, eventually leading to tracking failure. To solve this problem, we propose a simple and robust reliable memory model. In particular, an adaptive evaluation strategy (AES) is proposed to assess the reliability of tracking results. AES combines the confidence of the tracker predictions and the similarity distance, which is between the current predicted result and the existing tracking results. Based on the reliable results of AES selection, we designed an active–frozen memory model to store reliable results. Training samples stored in active memory are used to update the tracker, while frozen memory temporarily stores inactive samples. The active–frozen memory model maintains the diversity of samples while satisfying the limitation of storage. We performed comprehensive experiments on five benchmarks: OTB-2013, OTB-2015, UAV123, Temple-color-128, and VOT2016. The experimental results show that our tracker achieves state-of-the-art performance.

**Keywords:** online update; reliable evaluation strategy; active–frozen memory model; visual tracking

**Citation:** Ge, D.; Liu, R.; Li, Y.; Miao, G. Reliable Memory Model for Visual Tracking. *Electronics* **2021**, *10*, 2488. https://doi.org/10.3390/electronics10202488

Academic Editor: Christos J. Bouras

Received: 5 September 2021
Accepted: 30 September 2021
Published: 13 October 2021

## 1. Introduction

Visual tracking is a fundamental problem of computer vision that tracks targets in subsequent frames by specifying the position and size of the target in the first frame. It has been successfully applied to robots, video surveillance, and self-driving cars. There are some challenging factors, such as deformation, in-of-plane scale variation, and illumination variation. These challenges are likely to cause significant changes to the appearance of the target. Therefore, how to effectively learn the appearance change of a target is an essential issue of visual tracking.

Recently, online learning-based trackers have achieved good performance. Online updates are often employed to learn appearance changes of targets. The tracking results are collected as online training samples every frame or at fixed intervals. There are some online update strategies that have been proposed [1–6]. For example, some strategies include selecting the most confident tracking result within the fixed interval frames to update specific networks [7]; collecting two consecutive frames [2]; storing each frame in order [3,8,9]; using a convolutional neural network to update the template [4,5]; and storing all tracking results using the Gaussian Mixture Model (GMM) [1,10].

Although the functions of these online update strategies have been validated, there are still two challenges. One challenge is that tracking results are not always reliable. When misalignment, occlusion, and out-of-view occur, the tracking results are likely to contain a great amount of background information, which is regarded as noise. Unreliable tracking results reduce the ability of a tracker to distinguish between targets and backgrounds, ultimately leading to tracking failure. Another challenge is that tracking results are not appropriately stored. The predicted tracking result in each frame [8,9] or several tracking results with higher confidence [2,7] are stored. However, in these methods, there are very few online tracking samples and also only represent the latest appearance change of the target. This can easily cause the tracker to over-fit the current appearance of target.

To solve the above challenges, we propose a robust reliable memory model that can accurately evaluate the reliability of tracking results and efficiently store all reliable results. First, we propose an adaptive evaluation strategy (AES) to assess the reliability of tracking results. AES calculates the reliability weight based on the tracking confidence of the tracker prediction and the similarity distance, which is between the current predicted result and the existing tracking results. Reliability thresholds are adaptively calculated to enhance the generalization of AES. Only reliable tracking results were selected to construct online training samples. Based on the reliable results of the AES selection, inspired by the computer storage structure, we devised an active–frozen memory model to store all reliable tracking results. Training samples stored in active memory are used to update trackers online. The frozen memory temporarily stores some of the oldest results. The active–frozen memory model maintains the diversity of training samples by exchanging samples in two memories. Combined AES and the active–frozen memory model can effectively avoid introducing background information, while avoiding tracker over-fitting to the current target appearance.

The contributions are summarized as follows:

1. We propose an adaptive evaluation strategy (AES) for the reliability of tracking results. The AES adaptively calculates the reliability threshold $r$ by combining the similarity distance and the confidence of the tracker prediction to reduce the introduction of background information. It ensures the quality of online training samples to avoid bad online updates.
2. We propose an active–frozen memory model to efficiently store all reliable tracking results. Samples stored in active memory are used to update the tracker. The frozen memory stores some of the oldest samples. Samples exchange between the active memory and frozen memory to ensure the diversity of samples within the active memory. The active–frozen memory model avoids tracker over-fitting to current appearance changes.
3. We evaluate our proposed tracker on five benchmark datasets: OTB-2013, OTB-2015, UAV123, Temple-Color-128, and VOT2016. Our tracker obtains a 69.4 AUC score on OTB-2015. Experimental results show that our proposed tracking algorithm achieves state-of-the-art performance.

## 2. Related Work

When scale variability, deformation, and rotation occur, the appearance of the target tends to change significantly. How to effectively learn the appearance change of a target is an essential issue of visual tracking. Recently, most approaches utilize the tracking results as online training samples to fine-tune the tracker to learn the appearance change of targets.

**Reliability evaluation of tracking results.** The reliability of online training samples is key to update the tracker. There are two main strategies for constructing online training samples. One strategy is to directly use the tracking results as an online training sample, regardless of its reliability. Some trackers [1,2,8,10] collect one training sample based on the tracking result in each frame. Other trackers [3,11,12] draw in some positive and negative samples around the predicted target location. When tracking drift occurs, the tracking results are likely to contain a great amount of background information that contaminates the online training samples.

The second strategy is to only consider the confidence of the tracking results, which is predicted by the tracker. FCNT [7] collects the most confident tracking results within the intervening frames. STCT [13] sets a confidence threshold and collects the tracking results with a confidence higher than the threshold. However, the tracking results are predicted by the tracker, which is always more confident about its own predictions. Thus, incorrect tracking results are still likely to achieve high confidence. Different from the above methods, we designed a robust adaptive evaluation strategy (AES) to assess the reliability of the tracking results. The AES not only considers the confidence of the tracking results but also considers the similarity distance between the current predicted result and the existing tracking results.

**Storage of online training samples.** Existing trackers construct a fixed volume of space to store online training samples. Some trackers [2,7,13] maintain a very small space, which only store one or two samples, to reduce the amount of computation. CREST [2] stores only two samples, namely the last two frames. FCNT [7] stores only one training sample within the intervening frames. STCT [13] stores the tracking result with the confidence of the tracker prediction higher than a predefined threshold. These methods only collect a small amount of tracking results, making the tracker over-fit easily to the current training samples.

Other trackers collect large amounts of tracking results in large spaces. Some positive and negative samples are stored in each frame [3,11,12]. One sample is added in each frame [8,10]. UpdateNet [4] uses the initial frame and accumulated template to estimate the optimal template for the next frame. Meta-updater [6] integrates geometric, appearance, and discriminative cues to sequential information. In particular, ECO [1] employs the Gaussian Mixture Model (GMM) to reduce the redundancy of the training samples. When the number of samples reaches the maximum capacity, the tracker discards the oldest samples, which easily causes the tracker to over-fit to the current appearance of the target. We propose an active–frozen memory model to store all reliable tracking results. The training samples stored in the active memory are used to fine-tune the tracker. The frozen memory temporarily stores the sample, whose weight is less than a threshold, as discarded the by active memory. The samples in the active memory and frozen memory are exchanged to ensure the diversity of samples in the active memory.

## 3. Our Approach

As mentioned earlier, the reliability of training samples is very important for the online updating of a tracker. When occlusion and tracking misalignment occur, the tracking result has a good chance to contain background information, which can be regarded as noise. When the tracker is updated with these tracking results, the ability of the tracker to distinguish between the background and the target is reduced, and eventually it can lead to poor location estimation or tracking failure. As shown in Figure 1, ECO (red box) does not consider the reliability of the tracking result and is easily affected by similar objects, scale variables, and rotation. Our approach (green box) evaluates the reliability of the result to avoid introducing noise for generating better prediction results. As we know, the reliability of tracking results is not enough of a concern for researches. We obtained two observations by analyzing the confidence of the current tracking result and the similarity distance, which is between the current predicted result and the existing tracking results. Based on the two observations, an adaptive evaluation strategy (AES) was designed to evaluate the reliability of the tracking results.

**The first observation.** The similarity distance between the current predicted result and the existing tracking results increases significantly when tracking drift occurs. Figure 2 shows the change of the minimum distance during the tracking process. Around the 70th frame, the target jumps, causing the appearance to significantly change and leading to the similarity distance to increase rapidly. Thus, the similarity distance can help to recognize when the tracking drift occurs.

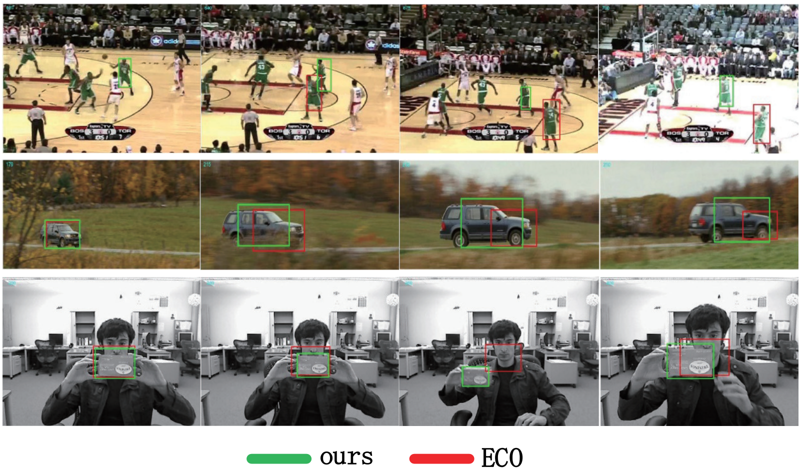


**Figure 1.** Our approach is compared with the ECO [1] on three test sequences called Basketball (**top row**), CarScale (**middle row**), and Twinnings (**bottom row**). ECO (red box) does not consider the reliability of the tracking result and is easily affected by similar objects, scale variability, and rotation. Our approach (green box) evaluates the result's reliability by AES.

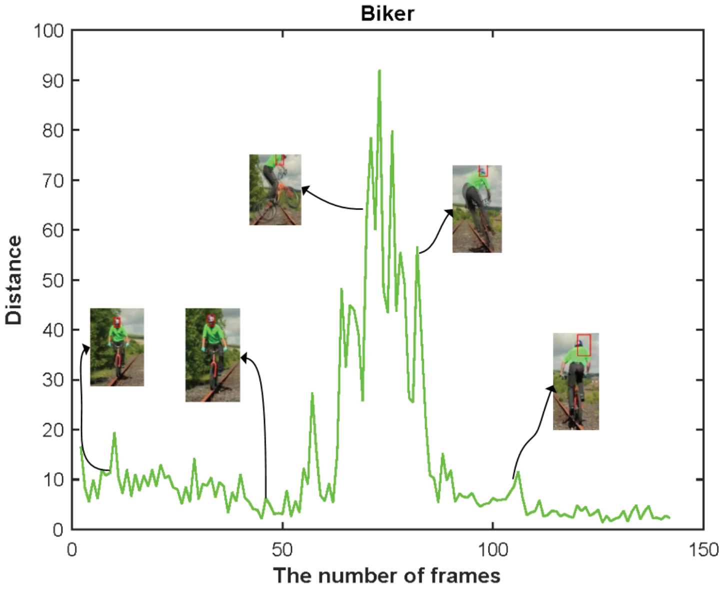


**Figure 2.** Visualization of the dynamic changes of the similarity distance on Biker. The tracking drift occurs when the target jumps around 70th frame. We can clearly observe that the similarity distance is significantly increased.

**The second observation.** We used the VGG network to extract semantic features and to represent the target with HOG and color name (CN) features together. The tracker has the ability to address some variations in the appearance of the target. Figure 3 shows the relationship of the similarity distance and the confidence. According to the first observation, as indicated by the purple curve, when the illumination or appearance of a target changes drastically, the similarity distance increases significantly. However, the confidence of the current predicted result (blue curve) is still higher than the mean confidence (red curve). That is, when the appearance of the target changes significantly, the tracker can still show a high level of confidence in the current prediction results.

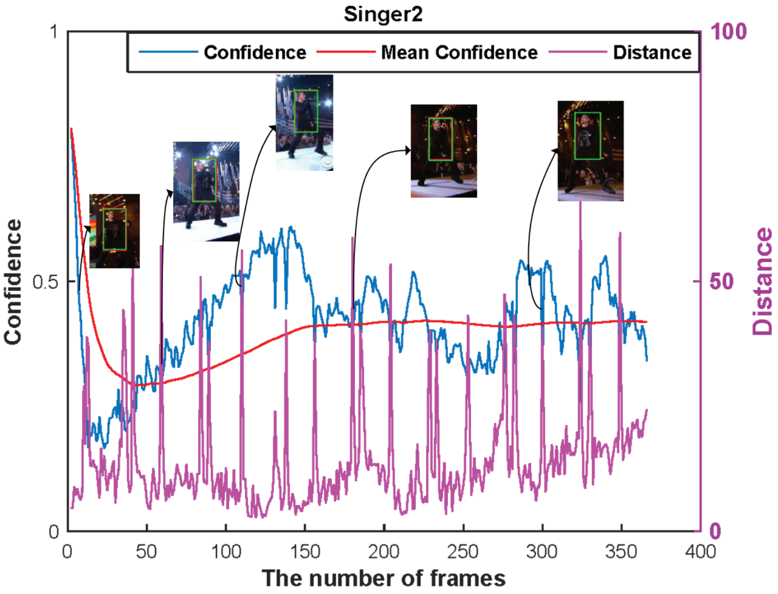


**Figure 3.** Visualization of the relationship between the confidence and similarity distance on Singer2. Even if the target's pose or appearance changes significantly (purple curve), the confidence of the current predicted result (blue curve) is still higher than the mean confidence (red curve).

The tracking results are collected as training samples to update the tracker online. Based on the reliable results of the AES selection, we designed an active–frozen memory model to maintain the diversity of results while satisfying the limitation of storage.

### *3.1. Adaptive Evaluation Strategy (AES) of the Reliability*

Inspired by the aforementioned two observations, we propose an adaptive evaluation strategy (AES) that combines the similarity distance with the confidence of the tracker prediction to assess the reliability of tracking results.

We use $U = \{u_1, ..., u_n\} \in R^{m*n}$ to represent the features of tracking results and $C = \{c_1, ..., c_n\} \in R^{1*n}$ to represent the confidence of the tracker prediction. For the current predicted result $x$, its tracking confidence is represented by $t$ and its reliable weight is represented by $V$. $V$ is composed of distance-based reliability weight $V_1$ and a confidence-based reliability weight $V_2$. When the current predicted result $x$ is unreliable, the $V$ is assigned a value of zero.

We first calculated the distance-based reliability weight $V_1$ based on the similarity distance between the current predicted result and the existing tracking results.

$$\underset{V_1}{min} E(V_1; r) = V_1 * \left( r - min \sum_{i=1}^{n} L(x, u_i) \right) \qquad s.t. \quad V_1 \in \{0, 1\} \tag{1}$$

where $L(x, y)$ calculates the Euclidean distance and $r$ is a threshold when $L(x, y)$ is greater than $r$, $V_1 = 0$, and otherwise is $V_1 = 1$. The purpose of $V_1$ is to help the tracker to identify significant changes in the appearance of the target. The confidence-based reliability weight $V_2$ is calculated according to the confidence of the tracking results.

$$\min_{V_2} E(V_2) = V_2 * \left( \frac{1}{n} \sum_{i=1}^{n} c_i - t \right) \qquad s.t. \quad V_2 \in \{0,1\} \tag{2}$$

The tracker is robust to appearance changes of the target because of the confidence-based reliability weight $V_2$. Based on the distance-based reliability weight $V_1$ and the confidence-based reliability weight $V_2$, the reliability weight $V$ is calculated by Equation (3).

$$V = V_1 \circ V_2 \tag{3}$$

where ∘ is a Hadamard product. According to Equations (1)–(3), the global optimum $V^\star$ of reliability weight $V$ is calculated by the following:

$$V^\star = \begin{cases} 1, V_1 \circ V_2 = 1 \\ 0, otherwise \end{cases} \tag{4}$$

The reliability of the tracking results can be effectively evaluated by Equation (4). The parameter $r$ is an important threshold that determines the reliability of the current predicted result. Figure 4 shows the similarity distance between the current predicted result and the existing tracking results in different sequences. In the FleetFace sequence (yellow curve), the similarity distance is significantly smaller than the Bolt2 sequence (red curve) and BlurCar1 sequence (green curve). In the Bolt2 sequence, the similarity distance shows a significant dynamic change. The similarity distance of different sequences is remarkably different because the target has different motion states, appearance changes, and resolutions of features. According to the second observation, the confidence of the tracking result can effectively address the appearance's change of the target. We propose a method that adaptively calculates the threshold $r$.

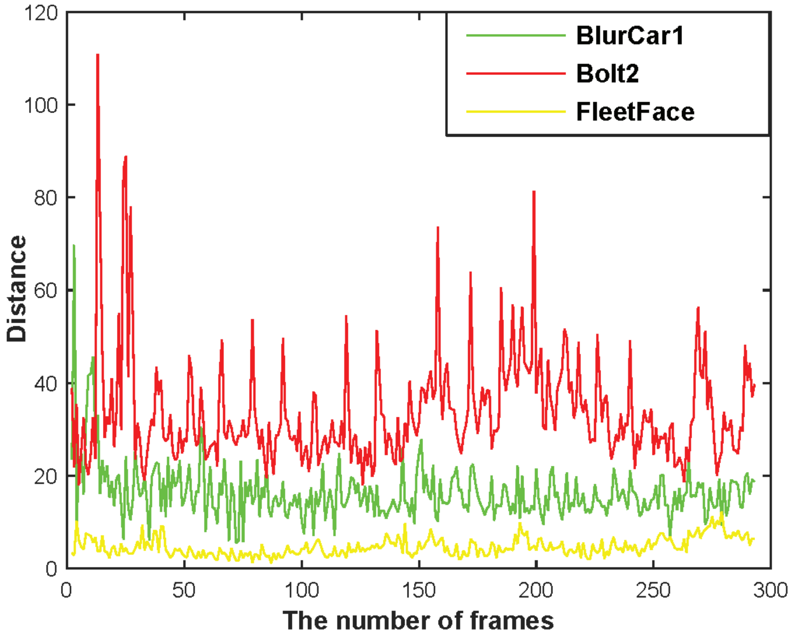


**Figure 4.** Visualization of the similarity distance between the current predicted result and the existing tracking results in the BlurCar1 (green curve), Bolt2 (red curve), and FleetFace (yellow curve). The similarity distance of different sequences is remarkably different.

In the case of $V_1 \bigoplus V_2 = 1$, this indicates that the distance-based reliability weight $V_1$ is different from the confidence-based reliability weight $V_2$. When $V_2 = 1$, this indicates that the appearance of the target has changed significantly. The threshold $r$ should be increased to select more tracking results as online tracking samples. When $V_2 = 0$, this indicates that the tracker is not certain about its own predictions. Although the new tracking results are close enough to the existing tracking results, we believe that the threshold $r$ should be reduced to ensure the quality of the current predicted result. The threshold $r$ can be adaptively calculated by the following formula:

$$r = r + w * \left[ r - min \sum_{i=1}^{n} L(x, u_i) \right] * \left( V_1 \bigoplus V_2 \right) \tag{5}$$

where $w$ represents the pace for each calculation.

### 3.2. Active–Frozen Memory Model

In order to learn the appearance change of the target, tracking results are collected as training samples to update the tracker online. Most trackers [1,7,10] discard the oldest results when the number of samples reaches the maximum limit, which results in training samples that do not fully represent the appearance change of the target.

Based on reliable results of the AES selection and as inspired by the multi-level cache technique in computer storage, we propose an active–frozen memory model that stores all reliable tracking results. The structure of the active–frozen memory model is shown in Figure 5, and is a cascaded structure that can exchange components between two memories. Tracking results stored in the active memory are used to update the tracker online. Frozen memory is used to temporarily store some of the oldest results. In order to reduce computation load, following the [1], we used the Gaussian Mixture Model (GMM) to fuse tracking results in each memory. The two closest components, namely K and S in GMM, are merged into one, specifically component G.

$$W_G = W_K + W_S, \quad \overline{X_G} = \frac{W_K \overline{X_K} + W_S \overline{X_S}}{W_K + W_S} \tag{6}$$

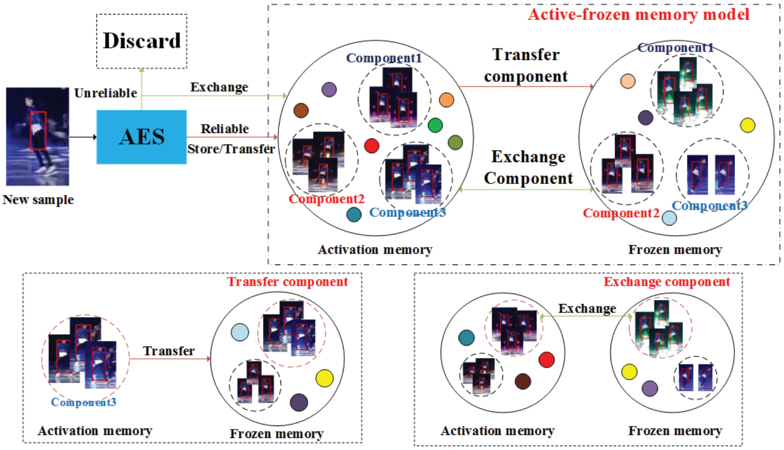


**Figure 5.** The structure of the active–frozen memory model (**top row**). There are two operations (**below row**), namely transfer component and exchange component. Only reliable tracking results are stored and are otherwise discarded directly. The active–frozen memory model guarantees the diversity and reliability of tracking results in active memory by exchange operations and AES.

We first constructed a Gaussian component based on the weight $W_x$ and mean features $\overline{X}$ of the current predicted result $x$. The reliability of $x$ was evaluated by AES (see Section 3.1 for details). If the current predicted result $x$ is reliable, it is stored in the active memory. Otherwise, it is discarded directly.

After the current predicted results are collected, we checked whether the component numbers in the active memory had reached the maximum limit and whether the weight of one component was less than the predefined threshold. If an existing component satisfies the above requirement, it is exchanged with the closest component from the frozen memory. If the frozen memory is empty, we place this component directly into the frozen memory. The active–frozen memory model guarantees the diversity and reliability of tracking results in the active memory. The stored procedure of the active–frozen memory model is illustrated in Algorithm 1.

**Algorithm 1** Stored procedure of the active–frozen memory model.

```
Require: current predicted result $x$.
Ensure: active–frozen memory.
Construct a component based on the weight $W_x$ and mean $\overline{x}$ of the new sample
Calculate the reliability weight $V$ of the tracking result $x$ by AES
if $V = 1$ (the tracking result $x$ is reliable) then
    The tracking result $x$ is stored in the active memory by Equation (6)
else
    Discard the tracking result $X$ directly
end if
if the number of components in the active memory reaches the maximum limit and
  one component with the weight is less than the threshold then
    if the frozen memory is empty then
        Put the component into the frozen memory directly
    else
        Exchange with the closest component from the frozen memory
    end if
end if
return active–frozen memory.
```

### 3.3. Model Update

In recent trackers [1,7,12,14], a sparse update scheme was employed. The tracker, which takes collected tracking results as online training samples, is updated every $N_s$ frames and each update performs a fixed number $N_i$ of iteration optimization algorithms. The sparse update scheme not only reduces the computations but also reduces the over-fitting to the recent online training samples.

We also utilized the sparse update scheme in our approach. Only the training samples stored in the active memory were used to update our tracker (see Section 3.2 for details). When the current predicted result was unreliable, the active memory did not change because the predicted result was discarded directly. Thus, before updating the tracker, we detected whether the active memory changes in the $N_s$ frame, that is, whether there were new tracking results to be collected. If the active memory had not changed, indicating the $N_s$ tracking results were unreliable, we reduced the number of iterations $N_i$ of the optimization algorithms to avoid the tracker over-fitting to existing online training samples. Otherwise, we performed $N_i$ times of iteration optimization algorithms.

## 4. Experiments

We validated the performance of our tracker on five benchmark datasets, including OTB-2013 [15], OTB-2015 [16], UVA123 [17], Temple-color-128 [18], and VOT2016 [19].

### *4.1. Implementation Details*

Our tracker was implemented in Pytorch. We initialized our tracker using the method proposed in [1]. The VGG-m network was used as a feature extractor to capture the $Conv1$ (the first convolutional layer) and $Conv5$ (the last convolutional layer) features, and the HOG and Color Name (CN) features were combined to represent the target. For the adaptive evaluation strategy (AES) of the reliability, the threshold $r$ was initialized to 0. In order to obtain a reasonable value of $r$, the tracking results of the first 50 frames were used to adaptively calculate the value of $r$ by Equation (5). In fact, the initial value of $r$ had no effect on the performance of the tracker. In the first 50 frames, the pace for each calculation $w$ was set to 0.5. In the subsequent frames, the pace $w$ was calculated by the following formula.

$$w = \begin{cases} 0.4 * max(c_i) + 0.6 * \frac{r}{distance_{min}}, & r > distance_{min} \\ 0.4 * max(c_i) + 0.6 * \left(\frac{r}{distance_{min}} - 1\right), & otherwise \end{cases} \tag{7}$$

where $distance_{min}$ represents the minimum similarity distance between the current predicted result and the existing online training samples.

For the active–frozen memory model, as presented in Section 3.2, the maximum limit of the number of training samples in the active memory and frozen memory was set to 50 and 10, respectively. We initialized the active memory with the tracking results of the first 50 frames of the sequence. The learning rate was set to 0.009. We updated the tracker every $N_s = 6$ frames. When tracking results were added to the active memory, we used the same iteration number $N_i = 5$ as in [1]. Conversely, the number of iterations $N_i$ was set to 4. Note that all parameters settings were kept fixed for all the sequences in the dataset. It is important to note that the computational complexity of our proposed adaptive evaluation strategy (AES) and active–frozen memory model was O(n), which is negligible and thus guarantees the real-time performance of the tracking.

### *4.2. Ablative Study*

In this section, we analyze the contribution of both the adaptive evaluation strategy (AES) of the reliability and the active–frozen memory model to the tracker by performing experiments on the OTB-2013 dataset [15]. The OTB-2013 dataset contains 50 sequences that are all fully annotated. There are 11 attributes, such as occlusion, scale transformation, and deformation, which represent the challenge factors in visual tracking. Each sequence has at least one challenge factor. We used a precision plot and a success plot to evaluate the performance of the tracker. Precision plots calculate the Euclidean distance between the estimated location and the ground truth, and counts the percentage of frames that are less than a given threshold distance. The threshold was set to 20 pixels. The success plot quantitatively calculates the overlap ratio of the bounding box, where the overlap rate ranges from 0 to 1. The success plot counts the number of frames whose overlap rate is greater than a given threshold. The threshold was set to 0.5.

We chose ECO [1] as our baseline tracker and organized four comparison experiments by controlling variables, including standard ECO, only the adaptive e- valuation strategy (ours-AES), only the active–frozen memory model (ours-AF memory), and our proposed approach (ours). Figure 6 shows the comparison experiment results on the OTB-2013 dataset. In the precision plot, the score of the baseline tracker was 93%. Compared with the baseline tracker, our active–frozen memory model achieved a 0.8% improvement and our adaptive evaluation strategy achieved a 1.6% improvement, which provided the greatest contribution. Our approach finally improved by 1.8%. In the success plot, the baseline tracker obtained an area-under-curve (AUC) score of 70.9%. Both the adaptive evaluation strategy and thw active–frozen memory model achieved a 0.4% improvement, and our approach achieved a 0.5% improvement compared with the baseline tracker.

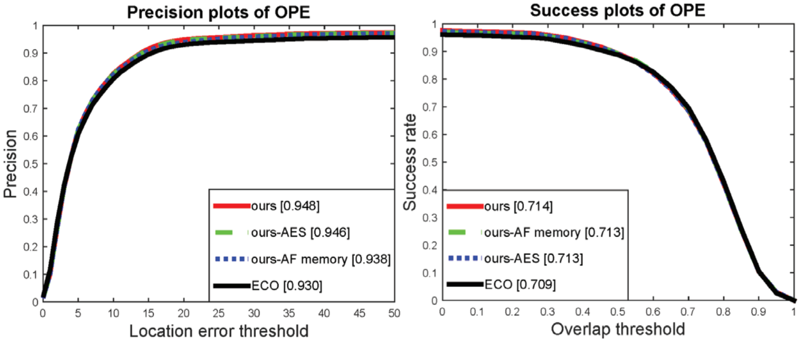


**Figure 6.** Ablative experiments on the OTB-2013 dataset. The area-under-curve (AUC) score of the success plot and the score of the precision plot are represented in the legend, respectively.

We also analyzed the performance of the tracker under different challenge factors. Figure 7 only shows the results of the scale variation, illumination variation, in-plane rotation, and deformation challenge factors; we achieved an increase of 1%, 1%, 0.6%, and 2.6% respectively. In particular, our method can better learn the deformation of a target, which is our main purpose, i.e., learning the appearance change of a target.

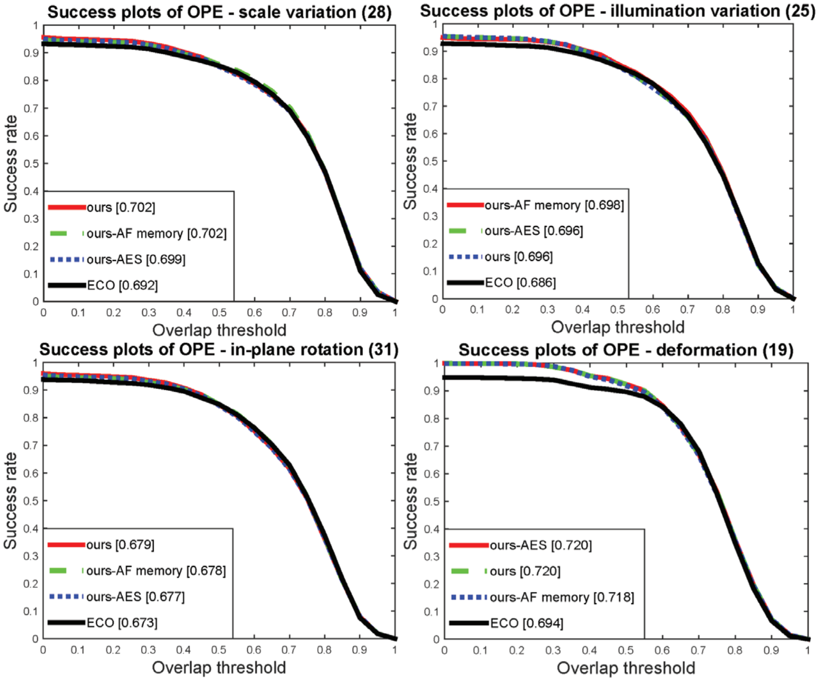


**Figure 7.** Success plot on scale variation, illumination variation, in-plane rotation, and deformation. The AUC score of each challenge factor is shown in the legend.

AES guarantees the quality of online training samples to avoid introducing background information and the active–frozen memory model guarantees the diversity of online training samples to prevent the tracker from over-fitting to the current target appearance. The experimental results in Figures 6 and 7 show that the adaptive evaluation strategy (AES) of the reliability and the active–frozen memory model are useful for improving the performance of the tracker.

Meanwhile, we conducted ablation experiments on VOT2016 [19] as shown in Table 1. Our tracker can reach 35 FPS with negligible computation introduced by AES and AF memory, satisfying the real-time requirement.

**Table 1.** Ablative experiments on the VOT2016.

| | Baseline | Ours-AES | Ours-AF Memory | Ours |
|---|---|---|---|---|
| EAO | 0.374 | 0.385 | 0.378 | 0.389 |
| A | 0.540 | 0.577 | 0.560 | 0.590 |
| R | 0.306 | 0.306 | 0.308 | 0.310 |
| FPS | 41 | 36 | 40 | 35 |

### *4.3. Comparisons to State-of-the-Art Trackers*

In this section, we compare our approach with state-of-the-art trackers on five benchmark datasets: OTB-2013 [15], OTB-2015 [16], UVA123 [17], Temple-color-128 [18], and VOT2016 [19].

**OTB-2013.** We compared our approach with VITAL [3], ECO [1], MDNET [12], DAT [11], MCPF [20], CREST [2], CCOT [9], TRACA [21], BACF [22], DeepSRDCF [23], SRDCF [8], SiamFC [24], and 29 trackers from the OTB-2013 dataset. The experimental results are shown in Figure 8. In the precision plot, VITAL achieved the best performance. Our tracker obtained a precision score of 94.8%, second only to VITAL and more than the 0.4% and 1.8% of DAT and ECO, respectively. In the success plot, our method achieved the best performance between all the state-of-the-art trackers, obtaining an AUC score of 71.4%, which was more than the 0.4% and 0.5% of VITAL and ECO, respectively. Compared with ECO, although the adaptive evaluation strategy (AES) of the reliability and the active–frozen memory model had been added, the extra calculations were negligible and our trackers ran at the same speed as ECO.

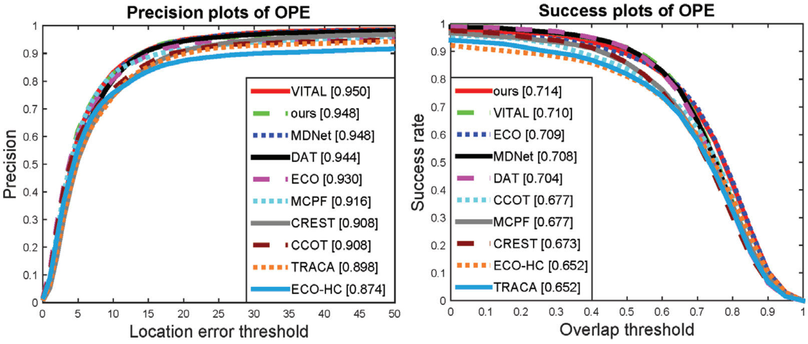


**Figure 8.** Precision plot and success plot on the OTB-2013 dataset. The AUV score and precision score of each tracker is shown in the legend. For clarity, we only show the top 10 trackers for performance.

**OTB-2015.** The OTB-2015 dataset is based on the OTB-2013 dataset, which adds 50 additional sequences and is still fully annotated. We compared our approach with recent state-of-the-art trackers: VITAL [3], ECO [1], MDNET [12], DAT [11], MCPF [20], CREST [2], CCOT [9], TRACA [21], BACF [22], DeepSRDCF [23], SRDCF [8], SiamFC [24], and 29 existing trackers from the OTB-2015 dataset. The experimental results are shown in Figure 9. Our approach achieved the best performance in both the precision and success plot, with a precision score of 92.3% and an AUC score of 69.4%, respectively. Our tracker was 0.5% higher than VITAL and 1.3% higher than VITAL in the precision plot. Additionally, our tracker was 0.3% higher than ECO and 1.2% higher than VITAL.

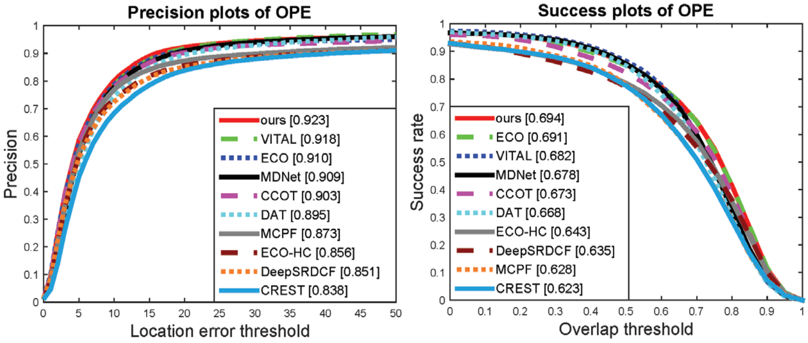


**Figure 9.** Precision plot and success plot for the OTB-2015 dataset. The AUV score and precision score of each tracker is shown in the legend. For clarity, we only show the top 10 trackers for performance.

**UAV123.** UAV123 is constructed by 123 video sequences and more than 110K frames, which contain 12 tracking attributes, captured from a low-altitude aerial perspective. We compared our approach with state-of-the-art trackers: ECO [1], MEEM [14], DSST [25], SRDCF [8], DCF [26], Struck [27], MUSTER [28], SAMF [29], and 31 trackers from the UAV123 dataset. Figure 10 shows the results over all the 123 sequences in the UAV123 dataset. Our tracker provided the best performance with a precision score of 74.9% and an AUC score of 52.8%. Additionally, our tracker achieved a substantial improvement over ECO [1], with a gain of 0.8% in the precision plot and a gain of 0.3% in the AUC.

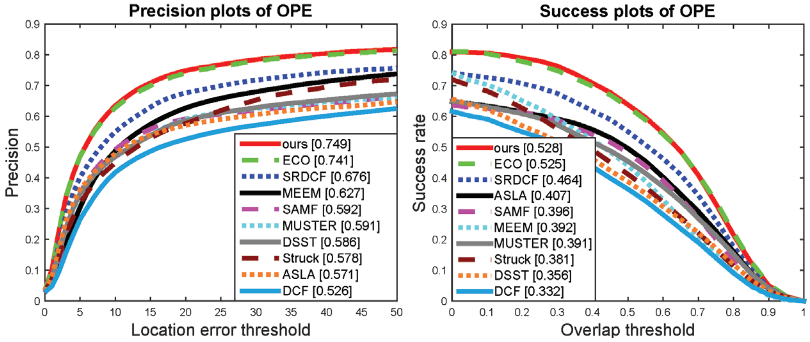


**Figure 10.** Precision plot and success plot on the UAV123 dataset. The AUV score and precision score of each tracker is shown in the legend. For clarity, we only show the top 10 trackers for performance.

**VOT2016.** The VOT2016 dataset contains 60 sequences with new annotations. We compared our approach with SiamDW [30], UpdateNet [4], SiamRPN [31], and ECO [1]. Table 2 shows the results of the VOT2016 dataset. Our tracker provided the best performance with an EAO score of 0.389.

**Table 2.** Comparison with state-of-the-art trackers on VOT2016.

| | SiamRPN | SiamDW-RPN | ECO | UpdateNet | Ours |
|---|---|---|---|---|---|
| EAO | 0.344 | 0.370 | 0.374 | 0.381 | 0.389 |
| A | 0.560 | 0.580 | 0.540 | 0.560 | 0.590 |
| R | 0.302 | 0.240 | 0.306 | 0.261 | 0.310 |
| FPS | 92 | 90 | 41 | 70 | 35 |

**Temple-color-128.** The Temple-color-128 dataset is constructed by 128 color sequences with ground truth and challenge factor annotations. As we all know, the color information of a target provides rich discriminative cues for inference. The purpose of this dataset was to study the use of color information for visual tracking. We compared our approach with MEEM [14], Struck [27], KCF [26], and other trackers from the Temple-color-128 dataset. The experimental results over all the sequences are shown in Figure 11. Our approach achieved the best performance in both the precision and success plot, with a precision score of 79.35.3% and an AUC score of 59.10%, respectively. Additionally, our tracker again achieved a substantial improvement over MEEM [14], with a gain of 8.54% in the precision plot and a gain of 9.10% in the AUC.

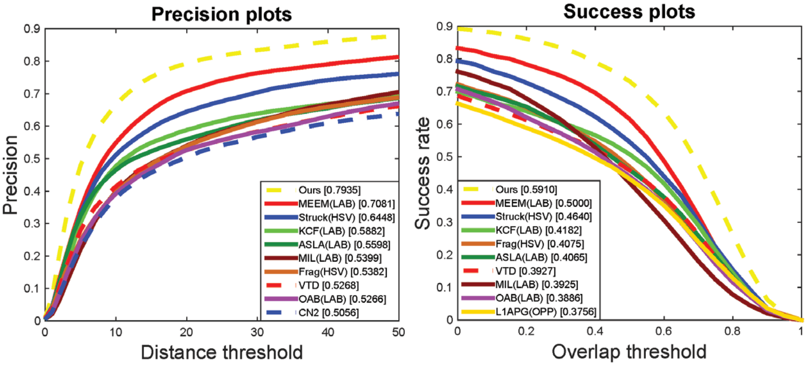


**Figure 11.** Precision and success plot on the Temple-color-128 dataset. The AUV and precision score of each tracker is shown in the legend. For clarity, we only show the top 10 trackers for performance.

## 5. Conclusions

In this paper, we proposed a robust strategy for constructing online training samples to learn the changes of a target's appearance. The adaptive evaluation strategy (AES) combines the tracking confidence of the tracker prediction and similarity distance, which is between the current predicted result and the existing tracking results, to assess the reliability of the tracking results in order to ensure the quality of the online training samples. We also proposed an active–frozen memory model that can effectively store all reliable tracking results. Training samples stored in the active memory are employed to update the tracker. The diversity of the online training samples is ensured by sample exchange between two memories to prevent the tracker from over-fitting to the current appearance changes. Extensive experiments on five benchmark datasets show that our approach outperforms the performance of state-of-the-art trackers.

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

**Funding:** The research study was jointly funded by the National Key R&D Program of China under grant number 2018YFC0807500; the National Natural Science Foundations of China under grant numbers 61772396, 61772392, 61902296, and 62002271; Xi'an Key Laboratory of Big Data and Intelligent Vision under grant number 201805053ZD4CG37; the National Natural Science Foundation of Shaanxi Province under grant number 2020JQ-330, 2020JM-195; the China Postdoctoral Science Foundation under grant number 2019M663640; and Guangxi Key Laboratory of Trusted Software (number KX202061); the Fundamental Research Funds for the Central Universities under grant No.XJS210310.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. Eco: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; Volume 1, p. 3.
2. Song, Y.; Ma, C.; Gong, L.; Zhang, J.; Lau, R.W.; Yang, M.-H. Crest: Convolutional residual learning for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2574–2583.
3. Song, Y.; Ma, C.; Wu, X.; Gong, L.; Bao, L.; Zuo, W.; Shen, C.; Lau, R.; Yang, M.-H. Vital: Visual tracking via adversarial learning. *arXiv* 2018, arXiv:1804.04273.
4. Zhang, L.; Gonzalez-Garcia, A.; Weijer, J.V.D.; Danelljan, M.; Khan, F.S. Learning the model update for siamese trackers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 4010–4019.
5. Li, P.; Chen, B.; Ouyang, W.; Wang, D.; Yang, X.; Lu, H. Gradnet: Gradient-guided network for visual object tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6162–6171.
6. Dai, K.; Zhang, Y.; Wang, D.; Li, J.; Lu, H.; Yang, X. High-performance long-term tracking with meta-updater. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6298–6307.
7. Wang, L.; Ouyang, W.; Wang, X.; Lu, H. Visual tracking with fully convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3119–3127.
8. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 4310–4318.
9. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 472–488.
10. Sun, C.; Wang, D.; Lu, H.; Yang, M.H. Correlation tracking via joint discrimination and reliability learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 489–497.
11. Pu, S.; Song, Y.; Ma, C.; Zhang, H.; Yang, M.H. Deep attentive tracking via reciprocative learning. *arXiv* **2018**, arXiv:1810.03851.
12. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2016; pp. 4293–4302.
13. Wang, L.; Ouyang, W.; Wang, X.; Lu, H. Stct: Sequentially training convolutional networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2016; pp. 1373–1381.
14. Zhang, J.; Ma, S.; Sclaroff, S. Meem: Robust tracking via multiple experts using entropy minimization. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 188–203.
15. Wu, Y.; Lim, J.; Yang, M.-H. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June2013; pp. 2411–2418.
16. Wu, Y.; Lim, J.; Yang, M.-H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 1834–1848. [CrossRef] [PubMed]
17. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking, In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 445–461.
18. Liang, P.; Blasch, E.; Ling, H. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5630–5644. [CrossRef] [PubMed]
19. Kristan, M.; Matas, J.; Leonardis, A.; Vojíř, T.; Pflugfelder, R.; Fernandez, G.; Nebehay, G.; Porikli, F.; Čehovin, L. A novel performance evaluation methodology for single-target trackers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2137–2155. [CrossRef] [PubMed]
20. Zhang, T.; Xu, C.; Yang, M.-H. Multi-task correlation particle filter for robust object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4335–4343.
21. Choi, J.; Chang, H.J.; Fischer, T.; Yun, S.; Lee, K.; Jeong, J.; Demiris, Y.; Choi, J.Y. Context-aware deep feature compression for high-speed visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June2018; pp. 479–488.
22. Kiani Galoogahi, H.; Fagg, A.; Lucey, S. Learning background-aware correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1135–1143.
23. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Convolutional features for correlation filter based visual tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 11–18 December 2015; pp. 58–66.
24. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 850–865.
25. Danelljan, M.; Häger, G.; Khan, F.; Felsberg, M. Accurate scale estimation for robust visual tracking. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014.

26. 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] [PubMed]
27. Hare, S.; Golodetz, S.; Saffari, A.; Vineet, V.; Cheng, M.M.; Hicks, S.L.; Torr, P.H. Struck: Structured output tracking with kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2096–2109. [CrossRef] [PubMed]
28. Hong, Z.; Chen, Z.; Wang, C.; Mei, X.; Prokhorov, D.; Tao, D. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 749–758.
29. Li, Y.; Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 8–11 September 2014; pp. 254–265.
30. Zhang, Z.; Peng, H. Deeper and wider siamese networks for real-time visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4591–4600.
31. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.

*electronics*

Article

# Efficient Opponent Exploitation in No-Limit Texas Hold'em Poker: A Neuroevolutionary Method Combined with Reinforcement Learning

**Jiahui Xu , Jing Chen and Shaofei Chen ***

College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410000, China; xjh@nudt.edu.cn (J.X.); chenjing01@nudt.edu.cn (J.C.)
* Correspondence: chenshaofei01@nudt.edu.cn

**Abstract:** In the development of artificial intelligence (AI), games have often served as benchmarks to promote remarkable breakthroughs in models and algorithms. No-limit Texas Hold'em (NLTH) is one of the most popular and challenging poker games. Despite numerous studies having been conducted on this subject, there are still some important problems that remain to be solved, such as opponent exploitation, which means to adaptively and effectively exploit specific opponent strategies; this is acknowledged as a vital issue especially in NLTH and many real-world scenarios. Previous researchers tried to use an off-policy reinforcement learning (RL) method to train agents that directly learn from historical strategy interactions but suffered from challenges of sparse rewards. Other researchers instead adopted neuroevolutionary (NE) method to replace RL for policy parameter updates but suffered from high sample complexity due to the large-scale problem of NLTH. In this work, we propose NE_RL, a novel method combing NE with RL for opponent exploitation in NLTH. Our method contains a hybrid framework that uses NE's advantage of evolutionary computation with a long-term fitness metric to address the sparse rewards feedback in NLTH and retains RL's gradient-based method for higher learning efficiency. Experimental results against multiple baseline opponents have proved the feasibility of our method with significant improvement compared to previous methods. We hope this paper provides an effective new approach for opponent exploitation in NLTH and other large-scale imperfect information games.

**Keywords:** opponent exploitation; no-limit Texas hold'em; neuroevolution; reinforcement learning

**Citation:** Xu, J.; Chen, J; Chen, S. Efficient Opponent Exploitation in No-Limit Texas Hold'em Poker: A Neuroevolutionary Method Combined with Reinforcement Learning. *Electronics* **2021**, *10*, 2087. https://doi.org/10.3390/electronics 10172087

Academic Editor: Amir Mosavi

Received: 20 July 2021
Accepted: 25 August 2021
Published: 28 August 2021

## 1. Introduction

Poker is often regarded as a representative problem for the branch of imperfect information games in game theory. It naturally and elegantly captures the challenges of hidden information for each private player [1]. The complexity of its solving method is much higher compared with perfect information games, such as Go [2]. As the most strategic and popular variation of poker, Texas Hold'em poker has been widely studied for years. AI researchers are working to find its solving method just as in AlphaGo or AlphaZero. However, Texas Hold'em poker contains additional challenges of imperfect information, dynamic decision-making, and misleading deceptions, as well as multistage chip and risk management, etc., which restrict it from being solved perfectly by AI. Most researchers are firmly convinced that the related technology behind the Texas Hold'em Poker's solution can be extended to multiple real-word applications, such as strategic portfolio, auction, finance, cybersecurity, and military applications [3], and the promising application prospect motivates continuous study until now.

Texas Hold 'em is an interactive decision poker game consisting of four stages: preflop, flop, turn, and river. At each stage players can bet different amounts of money based on private hands and public cards. They can only obtain rewards after taking a series of sequential actions until there is only one player remaining or the end of the last river stage. According to the limitation of betting amount, Texas Hold'em poker can be divided into either a limited game or a no-limit game. The number of their information sets are about $10^{14}$ and $10^{162}$, respectively [4]. It is obvious that solving no-limit Texas Hold'em (NLTH) is much more complex and resource-consuming, which makes NLTH an important benchmark in the domain of large-scale imperfect information games. The most recent and advanced progress of NLTH was achieved by two research teams from the University of Alberta (UoA), and Carnegie Mellon University (CMU). They have almost simultaneously put forward AI programs—Libratus (from CMU) [5] and DeepStack (from UoA) [6]—to solve two-player NLTH in 2017, as well as a Superhuman AI—Pluribus (from CMU) for multiplayer NLTH in 2019 [1].

Despite the above equilibrium-based solutions achieving highlight performances, their lack of adaptability to opponents seems to be a problem. That is, no matter what kind of opponents are there, these AIs always play the same way in order to ensure equilibrium. This is usually not the exact solution we want.

Another approach to deal with this problem is opponent exploitation . Simply put, opponent exploitation is a class of methods that specially design agents to target specific opponents. Sometimes, this approach can achieve greater rewards from the opponents than the equilibrium-based solutions, as it pursues the maximum individual utility against current opponent strategy rather than equalize all possible opponent strategies indiscriminately. Related research for opponent exploitation in NLTH can be seen in several previously published studies [7–12], using advanced techniques including deep reinforcement learning, neuroevolution, etc. While methods in these works show practical effectiveness, there still exist some disadvantages that need to be improved. For example, due to the multi-stage sequential decision process and high dimensional action/state space in NLTH, the reinforcement learning (RL) method typically confronts the challenge of sparse rewards that only obtain non-zero values at the final steps [9,10], whereas the neuroevolution (NE) method typically suffers from high sample complexity and struggles to optimize a large number of parameters [11,12]. Generally speaking, these problems can be summed up as ineffective and inefficient learning. For opponent exploitation in a large-scale problem such as NLTH, what we most want to achieve is not merely learning to exploit our opponents as much as possible (effectiveness). We also want to make the learning process as fast as possible (efficiency), which can greatly reduce the consumption of computing resources and time. Thus, developing an effective as well as efficient learning method for opponent exploitation in NLTH is the main motivation of our work.

In this paper, we propose a novel method combining neuroevolution (NE) with reinforcement learning (RL) for opponent exploitation in NLTH. The key insight of our method (NE_RL) is to incorporate NE's ability to address the challenge of sparse action reward in the RL framework by evaluating returns of entire game episodes (the amount of chips you win/lose) to form a fitness metric. Additionally, RL's ability to leverage powerful gradient descent methods can in turn help improve the learning efficiency in NE, which will greatly benefit the training process. In addition, NE_RL extends NE's population-based approach to build two separate populations that evolved by NE and RL seperately. Synchronous interactions within and between the populations can make the learning process more stable and robust. These improvements together make NE_RL a more effective and efficient opponent exploitation method compared to the previous NE- or RL-only methods. It should be noted that the NE method in this work refers to using evolutionary computation to optimize the weights of neural networks with fixed network topologies. The topologies or architectures are manually designed and improved, as discussed in Section 3.2.

## 2. Background

How AI can solve Texas Hold 'em poker has been a major challenge in recent years. One popular approach to achieve this goal is equilibrium-based solutions, which include the most part of state-of-the-art algorithms [2,5,6,13]. However, these leading "game-solving” paradigms still have some deficiencies in some aspects. For example, they need a large amount of computing resources to obtain so-called equilibrium solutions, and these equilibrium solutions do not take into account any advantage of opponents’ weakness that can be exploited, which corresponds to poor dynamic adaptiveness [14]. More importantly, the theoretical guarantees of the equilibrium no longer stand in multiplayer settings [15].

Basically speaking, the equilibrium solutions work under the assumption that the opponent is perfectly rational and then conduct a no-regret search or fictitious self-play along the entire (or compressed) game tree so as not to lose in expectation, no matter what the opponent does [16]. However, the assumption is makes it difficult to win often against specific opponents (either weak or skilled). Alternatively, we can consider one’s goal as learning to play and maximizing one’s rewards against some specific opponent groups through repeated strategic interactions (which is exactly the core of NLTH). In such a case, an equilibrium strategy is perhaps not so optimal and this is the problem that opponent exploitation mainly deals with. To illustrate at a high level, opponent exploitation means to win over one’s opponents as much as possible [17]. One possible approach to achieve this goal is to explicitly identify opponents’ hand beliefs [8] or strategy styles [18] and then make decisions accordingly. Similar methods can be collectively called “explicit exploitation” and are quite easy to understand. However, these types of methods rely very much on the accuracy of identification, which is as difficult (if not more) as solving the game itself and requires either sufficient domain knowledge or a mass of labeled data. Another possible approach called “implicit exploitation” seems to be more feasible [19]. Its main idea is to improve the rewards directly against the opponents through repeated interactions and a policy optimization function, which is similar to the idea of reinforcement learning. The biggest difference compared to “explicit exploitation” is that it does not explicitly reason about exploitable information about the opponents but instead implicitly learns to win through end-to-end training. Generally speaking, “implicit exploitation” is a more effective and straightforward class of opponent exploitation methods. The opponent exploitation methods mainly discussed in this paper all fall into this class.

In 2009, Nicolai and Hilderman first put forward the idea to use a neuroevolutionary (NE) method in NLTH [12]. Their NE agent was composed of 35-20-5 feed-forward neural networks, with a sigmoidal function applied at each level. However, their experimental results showed that the skill level of the evolved agents is limited, even though evolutionary heuristics such as co-evolution and halls of fame were used. In 2017’s AAAI conference, Xun Li presents a NE method to evolve adaptive LSTM (Long Short Term Memory Network) poker players featuring effective opponent exploitation [11]. His main contribution lies in the introduction of the LSTM to extract useful features and learn adaptive behaviors in NLTH. However, the use of the NE method needs a huge number of training episodes and a large amount of computing resources to update generation by generation, mainly due to high sample complexity facing a large-scale problem such as NLTH. Some researchers instead tried to use more sample-efficient reinforcement learning (RL) methods to train NLTH agents [9,10]. However, the multi-stage sequential decision process in NLTH with sparse rewards imposes restrictions on the policy gradients optimization function in many RL methods. As a result, efficient opponent exploitation in NLTH still remained a problem to be solved in recent years. Combining the NE method with the RL method is a general idea to complement each method’s flaws and use the strengths of both. In 1991, Ackley and Littman first showed the combination of evolutionary and gradient operators to be significantly more effective for agent survival in an uncertain environment [20]. In 1997, Chowdhury and Li used “messy genetic algorithms” to overcome the disadvantages of traditional RL techniques for fuzzy controllers [21]. In 2007, Lin et al. proposed R-HELA, a “reinforcement hybrid evolutionary learning algorithm”, for solving various control problems with uncertainty [22]. Later, in 2011, Koppejan and Whiteson presented the “neuroevolutionary reinforcement learning” method for generalized control of simulated helicopters and demonstrated that neuroevolution can be an effective tool for complex, online RL tasks [23]. More recently, similar ideas have been applied to some small-scale video games [24,25]. Nevertheless, applying such ideas in large-scale games such as NLTH has not been studied yet to the best of our knowledge.

In this work, we focus on the two-player and no-limit version of Texas Hold'em Poker, i.e., Heads-Up No-Limit Hold'em (HUNL). In HUNL, two players compete for money or chips contributed by both of them, i.e., the pot. At the beginning of each game, both players are forced to post a bet into the pot, i.e., the blinds. Then each player is dealt two cards, i.e., the hole cards, from a standard 52-card poker deck. The hole cards are private for the receiver, thus making the states of the game partially observable. The game is divided into four betting rounds: preflop, flop, turn, and river. The players act alternately in each betting round. Players must choose one of the following actions when it is their turn to act: call, check, raise, or fold. If a player chooses to call, that player will need to increase his/her bet until both players have the same number of chips. If one player raises, that player must first make up the chip difference and then place an additional bet. Check means that a player does not choose any action on the round but can only check if both players have the same chips. If a player chooses to fold, the game ends, and the other player wins the game. When all players have equal chips for the round, the game moves on to the next round. As the game proceeds, five cards are dealt face up on the table. Each of them is a community card, and the set of community cards is called the board. Specifically, the board is empty in preflop; three community cards are dealt at the beginning of the flop, a fourth community card is dealt at the beginning of the turn, and the last community card is dealt at the beginning of the river. The board is observable to both players throughout the game. If neither player has folded by the end of the river, the game goes into a showdown. In addition, at any point of the game, if a player who has moved all-in contributes no more chips to the pot than the other, the game also goes into a showdown. In this case, unfinished betting rounds are skipped, and the board is completed immediately. In a showdown, each player combines the hole cards with the board and chooses five out of the seven cards (two hole cards plus five community cards) to form the best possible hand. The player with the better hand wins the pot. If the two hands are equally strong, the players split the pot.

## 3. Methods

This paper introduces a new method, NE_RL, for opponent exploitation in NLTH. It incorporates NE's indifference to the sparsity of reward distribution and RL's gradient-based method to improve learning efficiency. Figure 1 illustrates the hybrid framework of NE_RL. It is divided into two parts by a dotted line. The left part shows a standard off-policy RL agents' training process and the right part is a standard neuroevolutionary agents' training process. The opponent pool contains a set of baseline opponent strategies for training. At the beginning, two populations of agents are initialized with different parameter distributions. Then, the individuals from the populations play full NLTH games against one (or more) specific opponent(s) sampled from the opponent pool separately. The populations are continuously updated by NE and RL respectively. The interactions between these two parts are arranged in the following ways (as shown by the red arrow): On the one hand, the trajectories generated by the NE agents can provide diverse experiences for the replay buffer to train the RL agents. On the other hand, we periodically use the RL agents to replace the worst performing NE agents to inject gradient information into the NE population (this process is abbreviated as transfer). There are several advantages to these interactions. First, the recycling of the NE agents' training data makes full use of information from each agent's experiences. Most samples can be used for further training rather than simply being discarded. As a result, the learning efficiency is improved with less game samples. Second, the gradient information provided by the RL agents can help guide the evolving directions for the NE population, which leads to faster learning compared to the NE-only method. Additionally, both the NE agents and the RL agents conduct population-based training, which aims to produce moderate diversity and redundancy. With diversity and redundancy, our method can explore the strategy space on a larger scale during the learning process. We argue that these two properties respond well in practice to the challenges of large-scale and high-complexity problems such as NLTH. Experimental results against multiple baseline opponents have proven the feasibility of our method with a significant improvement compared to the previous NE and RL methods.

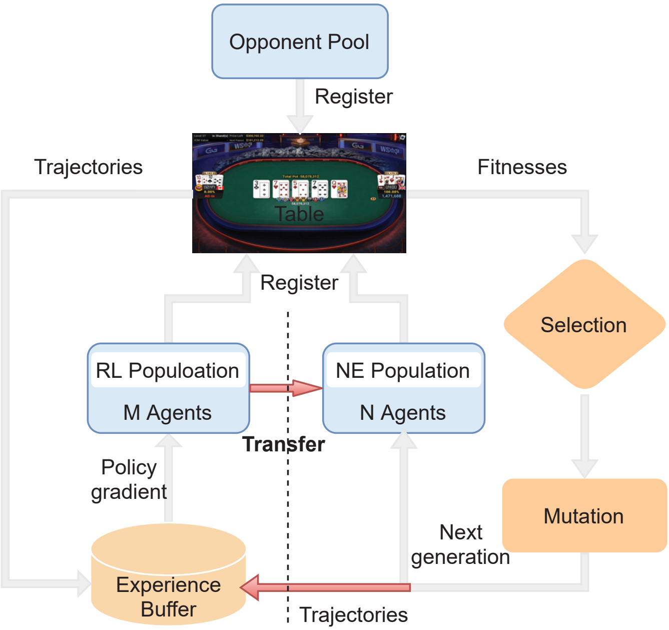


**Figure 1.** High-level illustration of the NE_RL method for opponent exploitation in NLTH.

Further details are described in the following subsections on two levels: learning methods and architectures. Section 3.1 separately introduces the learning methods (NE and RL) used in our work. Section 3.2 mainly describes how to organically combine these two learning methods together and then introduce the fundamental network architectures that the learning methods can apply to.

### *3.1. Introductions of NE and RL*

NE is a class of black box optimization algorithms typically used for neural network (NN)-based modules [26]. Inspired by natural evolution, the general flow of NE is as follows: At every iteration, a population of NN's parameter vectors is perturbed by mutation or a crossover operator and their objective function values ("fitness") are evaluated. The highest scoring parameter vectors are then recombined to form the population for the next iteration. There are various implementations of NE depending on the specific problem and context. In our method, in order to incorporate with RL we adopted a standard NE algorithm that proceeds as follows: A population of NE agents is initialized with random weights. They are then evaluated in interactions with the same opponent concurrently and independently. Each agent's cumulative rewards in the current generation are averaged to serve as its fitness. A selection operator then selects a portion of the population for survival with a probability commensurate with their relative fitness scores. The agents in the population are then probabilistically perturbed through mutation and crossover operations to create the next generation of agents. A select portion of agents with the highest relative fitness are preserved as elites and are shielded from the mutation step.

The RL population composed of multiple agents is initialized with different parameter distributions and updated independently by the same RL method. The RL method in this work typically refers to any off-policy reinforcement learner that utilizes a cyclic replay buffer $R$ maintained by both the RL agents and the NE agents. Trajectories in $R$ come from separate game episodes that contain a tuple ($s_t$,$a_t$,$r_t$,$s_{t+1}$), which refers to the current state, action, observed reward, and the next state, respectively. Here we take a Deep Q-learning Network (DQN) method as example to illustrate a representative learning process of RL. First, compute the current state's estimated Q-value via $Q(s, a; \theta_i)$ ($\theta_i$ refers to the parameters of the Q-networks). The corresponding loss function is:

$$L_i(\theta_i) = \mathbb{E}_{s,a \sim \rho(\cdot)}\left[(y_i - Q(s, a\ ; \theta_i))^2\right] \text{ where:}$$
$$y_i = \mathbb{E}_{s' \sim \mathcal{E}}[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) \mid s, a]$$

Then, use an optimizer such as Adam to minimize the loss function and perform a gradient descent update. Usually the policy of the DQN agent during training is a noisy version of the learned policy: $\pi_b(s_t) = \pi(s_t) + Noise(t)$, where $\pi(s_t) = argmaxQ(s_t, a_t)$ is a greedy policy over the estimated Q-value at time step $t$ and the value of $Noise(t)$ decays with $t$. The additional noise is for the purpose of exploration in the RL agents' action space.

### *3.2. Combination of NE and RL*

The main idea of our method, NE_RL, is using the advantages of the combination of NE and RL. Algorithm 1 provides a detailed pseudocode as well as the complete procedure of NE_RL. The size of the NE population ($N$), the size of the RL population ($M$), and the size of the replay buffer $R$ are important hyperparameters of the algorithm. $Gen_{max}$ refers to the max generations of algorithm updates. $f_{NE}$ and $f_{RL}$ are computed as each agent's fitness value via a Tournament function, which conducts $\xi$ full game episodes and returns averaged game results. The action–reward tuples of $(s_i, a_i, r_i, s_{i+1})$ are extracted from the game episodes and restored in the replay buffer so that the RL agents can continually learn from them. The RL agents use DQN to update their action-value function $Q$ with weights $\theta$ and $y_i$ represent the q-target value, which is then used along with the predicted q-eval value $Q(s, a\ ; \theta)$ to compute the mean square loss function $L_i(\theta_i)$. At first sight, one may find it similar to a standard NE method. Compared to NE, which just uses episodes to compute a fitness score, NE_RL also looks into the episodes to extract experience to learn. It stores both the NE agents' and the RL agents' experiences in its replay buffer $R$ rather than disregard them immediately. The RL agent can then sample a small batch from $R$ and use it to update its parameters by gradient descent. It is obvious that with this mechanism we can extract maximal information from each individual experience and improve the method's sample efficiency.

As the fitness score captures an individual's episode-wide return, the selection operator imposes strong pressure in favor of individuals with higher episode-wide returns. Since the replay buffer consists of the experiences gathered by these individuals, this process skews the state distribution towards regions with higher episode-wide returns. This is a form of implicit prioritization that favors an experience with a higher long-term return, and is effective for NLTH with long time horizons and sparse rewards. RL agents using this state distribution tend to learn strategies to achieve higher episode-wide returns. In addition, a noisy version of the RL agent is used to generate an additional experience for the replay buffer. In contrast to the population of agents that explore by noise in their parameter space (neural weights), the RL agents explore through noise in their action space. These two processes complement each other and form an effective exploration strategy to better explore the policy space.

The final procedure of NE_RL involves contributions from RL agents to NE population. Periodically, the RL agents' parameters are duplicated into the NE population; we define this process as transfer. It is the key process in which the NE population can directly receive the information learned through gradient descent. If the information from the RL agents is good, it will survive the selection operator and pass on to the NE population via the crossover operator. Otherwise, it will simply be discarded. Such a mechanism only allows constructive information to flow from the RL agents to the NE population.

**Algorithm 1** Main procedure of NE_RL

```
Initialize a population of $N$ NE agents $pop_n$ and an empty cyclic replay buffer $R$
Initialize a RL agent $A_{rl}$ with weights $\theta^{\pi}$ and make $M$ copies to form a RL population $pop_m$
Define a noise generator $O$
for generation from 1 to $Gen_{max}$ do
    for agent $A \in pop_n$ do
        $f_{NE}$, $R$ = Tournament($A, \xi, noise = None, R$)
    end for
    Rank the population based on fitness scores and conduct the selection, crossover, mutation operators respectively
    $f_{RL}$, $R$ = Tournament($A_{rl}, R, noise = O, \xi$)
    Sample a random minibatch of T transitions $(s_i, a_i, r_i, s_{i+1})$ from R
    Set $y_i = \begin{cases} r_i, & \text{for terminals}_{i+1}; \\ r_i + \gamma \max_{a'} Q(s_{i+1}, a'; \theta), & \text{for non-terminals}_{i+1} \end{cases}$
    Update $A_{rl}$ by minimizing the loss: $L_i(\theta_i) = \mathbb{E}_{s,a \sim \rho(\cdot)} \left[ (y_i - Q(s, a ; \theta))^2 \right]$
    Copy $pop_m$ into $pop_n$: for the $M$ weakest set $A_M \in pop_n : A_M \Leftarrow pop_m$
end for

function TOURNAMENT($A, \xi, noise = None, R$)
    Sample an agent $A_{opp}$ from the opponent pool
    Reset and register $A$, $A_{opp}$ into the table
    $fitness = 0$
    for i = 1:$\xi$ do
        Choose action from policy $a_t = A(s_t) + noise_t$
        Send action $a_t$ to game engine and receive reward $r_t$ as well as new state $s_{t+1}$
        Store transition $(s_t, a_t, r_t, s_{t+1})$ into $R$
        $fitness \leftarrow fitness + r_t$ and $s_t = s_{t+1}$
    end for
    Return $\frac{fitness}{\xi}$, $R$
end function
```

Up to now, we have provided a complete introduction of our NE_RL method. However, these introductions are all about the agents' learning method, that is, how to combine NE with RL at the method design level. In order to make such a hybrid learning method possible to implement, the fundamental network architecture of the RL agents and the NE agents should be the same. As shown in Figures 2 and 3, we introduce two architectures that are used in our work.

Figure 2 is a prototype architecture mainly designed for early experiments on the combination of the NE and RL methods (henceforth referred to Arc_pro). As seen in Arc_pro, it receives inputs containing two sets of domain-specific features. One (game feature encoding) is made up of detailed information about the current game in which the agent is playing, and the other (opponent feature encoding) is a global vector that tracks the overall performance (simply represented by frequency of actions) of the opponent against which the agent is playing. They are concatenated to form the input tensor and then transferred into hidden layers that consist of fully connected neural networks with the size 512-1024-2048-1024-512. Additionally, the output tensor is mapped to the size of the action space and transformed into an action-value vector. The action space is continuous and infinite in NLTH; however, we observed that it could encourage more exploration by discretizing the actions. This forced the actions to be non-smooth with respect to input observations and parameter perturbations, and thereby encouraged a wide variety of behaviors to be played out. Similar conclusions can be found in Salimans's work [27]. This architecture is quite straightforward and easily compatible with both the NE and RL methods. It is mainly used to carry out preliminary experiments to verify the feasibility of our method. In the next section we will show the experimental results obtained with Arc_pro to effectively train NLTH agents to exploit weak opponents, such a random agents (RA). Further introductions of game feature encoding rules are as follows:

- Current stage: The current street (preflop, flop, turn, or river) as a one-hot encoding. It costs four features. It has to be a one-hot encoding as there is no continuity between the streets and each has to be considered separately.
- Hand equity: The equity is the probability of winning the hand if all active players went to the showdown. It is the only information about the cards that the player gets. It is calculated by Monte-Carlo simulations. In each simulation a random hand is given to the opponent, and random community cards are added to have a full board. The winner of the hand is then determined. These results are then averaged to estimate the equity. The simulations are repeated until a satisfying standard deviation on the estimated equity is achieved. The tolerance is set to 0.1%. Note that the hand range of the opponent is not explicitly modeled here, as random cards are distributed to him. The equity calculation is the most time-consuming operation in the agent's decision making, and thus also the generation of game data is time-consuming. In the implementation used it took approximately 10 ms. The repeated estimation and comparison of hand strength is the reason for this relatively long computation.
- My investment: The investment of the player in the hand, normalized by the initial stack.
- Opp's investment: The investment of the opponent in the hand, normalized by the initial stack. The investments summarize the importance of the current hand and may describe the strength of the hand the opponent is representing.
- Pot odds: The pot odds are the amount of chips necessary to call divided by the total pot. It is an important measure often used by professional players.

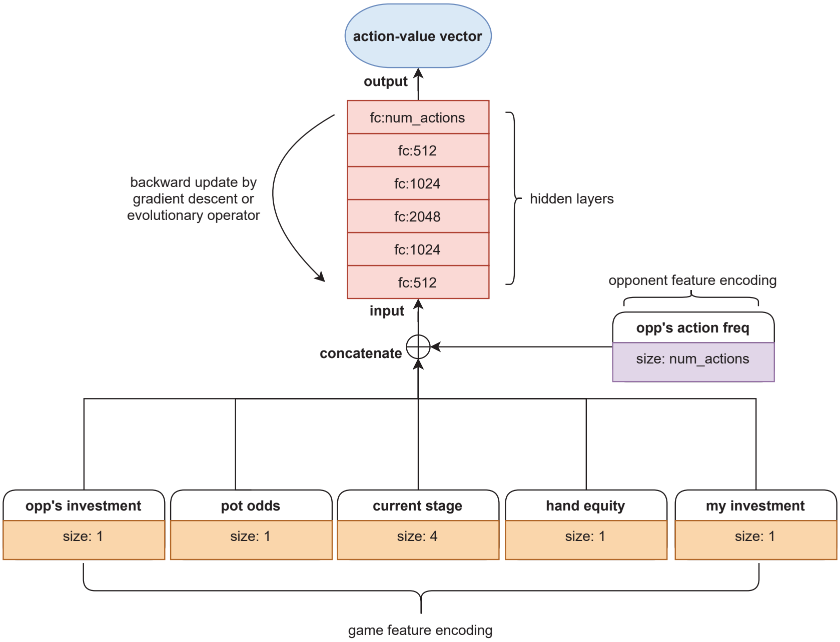


**Figure 2.** Arc_pro, a prototype architecture for the NE and RL agents. It receives domain-specific inputs and transfers to output action values with several layers of full-connected (fc) neural networks.

Figure 3 shows an improved architecture compared to Arc_pro. The biggest difference is the addition of an LSTM layer, a special class of recurrent neural networks that has been proven to be effective for dealing with sequential decision problems like NLTH. Since strategies in NLTH are essentially based on sequences of actions from different players, LSTM is directly applicable to extracting useful temporal features and learning adaptive behaviors. When we use this architecture (henceforth referred to as Arc_lstm) to train agents, the input features should be managed into episodic sequences. That means that the game trajectories are no longer considered independent but as interrelated within each episode. If an input feature implies the beginning of a game, the LSTM layer will reset to the initialization state in preparation for processing new episodic sequences. Therefore, the main advantage of Arc_lstm is that it captures the sequential nature of NLTH and makes decisions depending on the context of the current game stage as well as the opponent's overall performance (see the opponent feature set introduced above). It should be noted that in order to generate episodic continuous experiences, the replay buffer must be managed in sequence and the random minibatch sampled from it must contain full episodes rather than independent trajectories. This is another aspect that differs from Arc_pro. Experimental results show that these additional settings can help Arc_lstm achieve better performance against opponents whose strategy is relatively strong and harder to exploit.

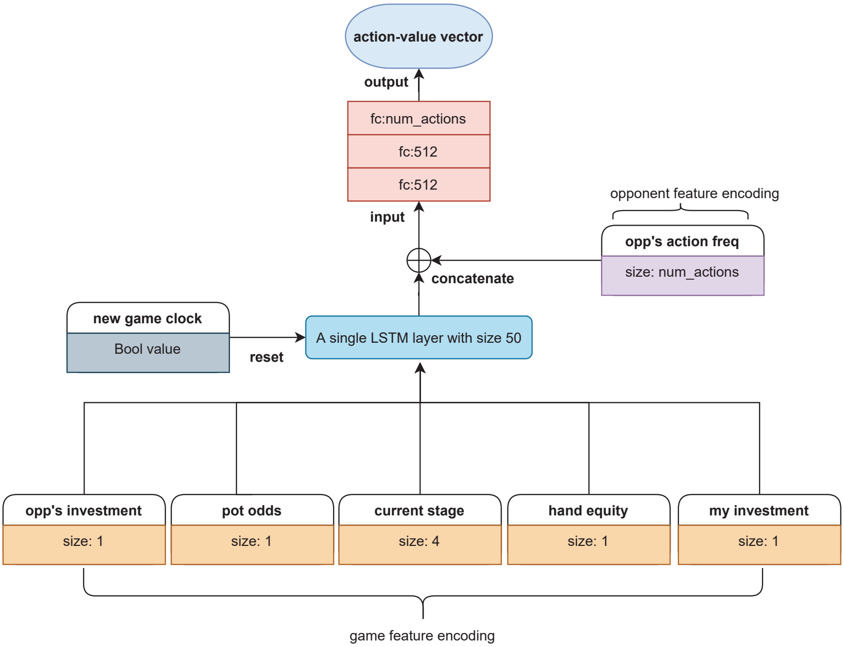


**Figure 3.** Arc_lstm, an improved architecture compared to Arc_pro. The biggest difference is the addition of an LSTM layer, which helps to extract useful temporal features and learning adaptive behaviors.

## 4. Experiments

In this section, we present experiments conducted to evaluate the proposed NE_RL method for NLTH. We first compared the performance of NE_RL with the NE- or RL-only methods when training NLTH agents against weak opponents, such as random agents. More specifically, these NLTH agents share the same architecture as Arc_pro but differ in their learning methods (respectively including an evolutionary method, a policy gradient, and the combination of both). With the feasibility of NE_RL validated, we stepped forward to train agents to exploit relatively strong opponents but encountered the problem that the NE method fails to learn anything after tens of thousands of episodes played, which also leads to a poor learning process for NE_RL. Then we used an improved architecture, Arc_lstm, leading to a significant performance boost, thus further validating the role of the additional LSTM in Arc_lstm. We then evaluated the performance of NE_RL with Arc_lstm under a constrained situation in which training episodes were limited to 40,000. Such a limitation is necessary to evaluate the efficiency of all of the learning methods from a practical point of view, because the computational resource consumption grows linearly with the total training episodes. Finally, we conducted ablation experiments to demonstrate the effectiveness of the transfer process, which is the core mechanism within NE_RL for incorporating evolutionary methods and policy gradient methods to achieve the best of both methods.

In all experiments, we chose the two-player NLTH setting in order to focus more on the method itself and leave the extension to multiplayer settings for future research. In addition, we used an open-source toolkit—Rlcard [4]—to carry out all experiments, so as to ensure reproduction of our results. All of the algorithms are based on PyTorch [28] and run through a cloud server with 80 CPUs. The duration of the experiments ranged from hours to tens of hours with multiple threads. The hyper-parameters of each algorithm and related experimental details are as follows:

- NE population size $N = 8$
- RL population size $M = 4$
- Size of replay buffer $R = 5 \times 10^4$
- Learning rate in DQN = $5 \times 10^{-5}$
- Discount rate in DQN = 0.99
- Training batch size in DQN = 32
- Elite fraction in NE = 0.25
- Mutation rate in NE = 0.3–0.05, mutation strength in NE = 0.5–0.1
  The mutation rate is set to linearly decrease from 0.3 to 0.05 while the mutation strength linearly is set to decrease from 0.5 to 0.1.
- Number of game episodes in tournament function $\xi = 2000$
  Since NLTH is a stochastic game with much uncertainty, 2000 independent full episodes were tested to compute an averaged fitness score.

### *4.1. Preliminary Evaluation of NE_RL*

The first experiment is a motivating example that validates the effectiveness of our method. In it we used a chump opponent random agent (RA) to train NLTH agents with the same architecture (Arc_pro) but different learning methods, and their performances were evaluated periodically after a certain number of episodes played. Figure 4 (Left) shows the comparative performances of methods in previous works (DQN, NE) and our proposed method, NE_RL, which combines the mechanisms within DQN and NE. The rewards of the agents were measured by the average winning big blinds per hand (bb/h for short, each player was initialized with 50 bb for the beginning of each hand). Higher rewards represent higher exploitation of the opponent. Specifically, the evaluation process ran every 100 training episodes for DQN and every evolutionary generation for NE and NE_RL. During the evaluation process, another 2000 episodes were played against the RA for each agent to compute an average reward, which was necessary in order to reduce the impact of luck and uncertainty in NLTH. This experiment was conducted fivr times to compute averaged performance with the standard error band. From the experimental result we can see that the DQN agent learned faster with much less episodes and converged at an exploitation level of about 8.5 bb/h, while the NE population typically needed many more interactions against their opponent but could converge to the exploitability bound computed by the local best response [29], owing to the population-based exploration. We were also excited to see that the NE_RL agents inherited the advantages from both methods and achieved better overall performance on sample efficiency as well as maximum exploitation. Maximum exploitation means the largest amount of money one can win from the opponent and sample efficiency means that the lowest number of training episodes needed to achieve converged performance. These together constitute the goal of our work and the motivation to propose the NE_RL method.

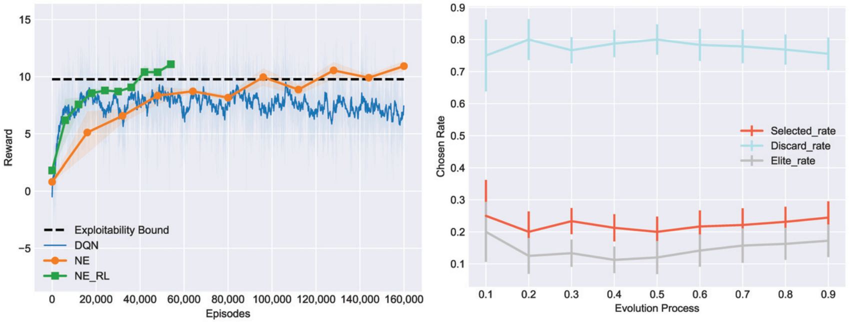


**Figure 4.** (**Left**) Comparative performance of NE_RL, NE, and DQN in training against the RA opponent. (**Right**) Average selection rate of the transferred RL agents changes during the course of evolution and plot with error bars.

This result validates the feasibility of NE_RL. It outperformed NE and RL by combining the advantages of both methods. The interactions between NE and RL include two types: the training episodes of the NE population are reused to provide diverse experience for the RL agents to train, while the trained RL agents periodically transfer to the NE population to inject gradient information into the NE agents. In order to examine whether such interactions truly help achieve improved performance, we ran additional experiments logging how the transferred RL agents performed among the NE population. Three evaluation indexes were used to log the frequency of these agents chosen as elites, selected, or discarded during selection: elite_rate, selected_rate, and discard_rate, respectively. As shown in Figure 4 (Right), during the course of evolution, the chosen rate of the transferred RL agents remained stable. elite_rate and selected_rate indicate that the gradient information played a role among the NE population and contributes to the evolutionary direction. In other words, the NE population may spend fewer generations to find its way to evolve with the help of gradient information, and the performance of the population continually improves though the discard_rate of transferred RL agents is high. This is due to built-in redundancies and diversities of population-based exploration and exploitation.

In conclusion, the first experiment successfully validated our NE_RL method's effectiveness for opponent exploitation in NLTH and achieved significant improved performance compared to previous NE- or RL-only methods. However, in this phase we only chose a weak opponent that plays randomly for testing purposes. Further experiments against stronger opponents will be introduced in the following subsection.

### *4.2. Learning to Exploit Baseline Opponents*

The goal of our work is proposing a general method for opponent exploitation in NLTH. The first experiment proved our method's feasibility in NLTH games against a weak opponent. In the next experiment, we introduced four stronger agents adopted from an open-source NLTH platform [30]. They were designed according to specific rules and characterized by human-like styles, namely Tight Aggressive (TA), Tight Passive (TP), Loose Aggressive (LA), and Loose Passive (LP). Unlike the RA opponent that plays randomly, these baseline opponents take actions based on their hand-strength whose strategies are relatively strong. "Tight" means an opponent only plays a small range of strong hands and "Loose" means the opposite. "Aggressive" means an offensive play style while "Passive" means a defensive play style. Moreover, the TA opponent's decision-making mechanism also mimics human bluffing behavior and probabilistically takes deceptive actions, which makes it even harder to exploit. When we tried to do the same thing to train agents against these baseline opponents as before, problems began to arise. We observed that the NE method seems to make no progress even after tens of thousands of training episodes. This led us to reflect on what went wrong, such as whether the network architecture Arc_pro needs improvements to adapt to stronger opponents. The main reason may be that the prototype structure of the neural network originally designed in Arc_pro is not available for generating strategies against more complex opponents.

Inspired by previous works that highlighted a special class of recurrent neural networks, LSTM (Long Short Term Memory), as an effective and scalable model for NLTH, we modified Arc_pro by adding a single LSTM layer between the input game feature set and the hidden layers. The main purpose was to extract the temporal features from opponents' actions, which were simply neglected in the first experiment, since the RA opponent just plays randomly and there is no temporal feature to extract at all. However, if we want to exploit more experienced opponents, these extra features seem to become critical. Figure 5 shows the comparative performance when we replace Arc_pro with the improved architecture Arc_lstm and use the same NE method to train agents against the baseline opponents (take TP for example). From the results we can see that the NE method recovered to optimize its performance once equipped with Arc_lstm. We can now conclude that Arc_lstm is a more suitable architecture to train NLTH agents. This is a crucial improvement not only for the NE method itself but also for our NE_RL method, since the NE method is the most important component of our NE_RL method. In the following experiments we continued testing the feasibility of our NE_RL method with Arc_lstm.

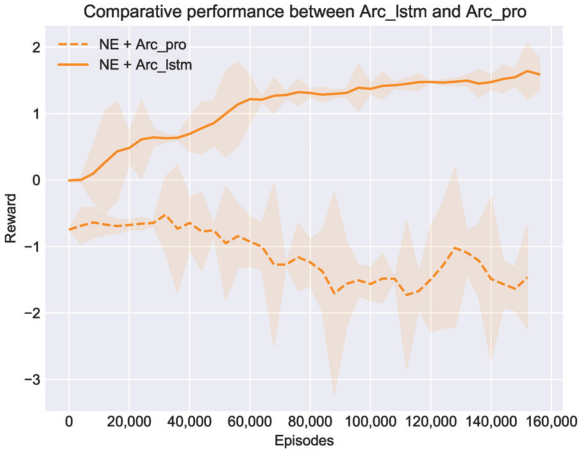
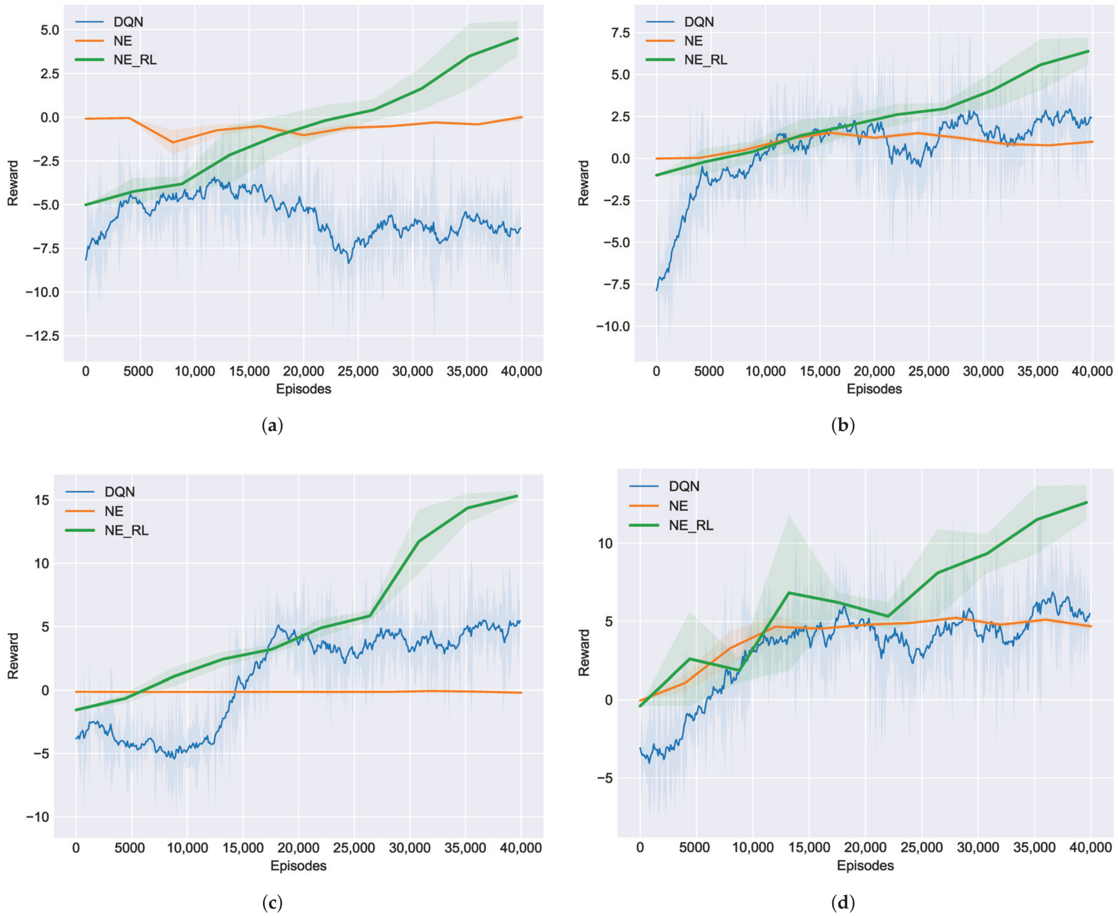


**Figure 5.** Comparative performance between Arc_pro and Arc_lstm with the same NE method in training against the TP opponent.

As we have emphasized in this work and validated in the first experiment, the NE_RL method is supposed to be superior to the previous NE- and RL-only methods for opponent exploitation in NLTH. With Arc_lstm we can now compare the performance between these methods against the four baseline opponents to show more evidence of the superiority of our method. As shown in Figure 6, we conducted four independent experiments against each of the baseline opponents (TA, TP, LA, LP) to evaluate the performance of opponent exploitation under different methods. Additionally, we set the maximum number of training episodes to 40,000 to reflect both the sample efficiency and the exploitation performance for each method. Together, these results suggest that our NE_RL method can achieve maximum exploitation of the baseline opponents under limited training episodes. By contrast, the NE method suffered from a lack of samples and the RL method could only converge to a sub-optimal performance under the same conditions. Similarly to the first experiment, we further studied the role of interactions between the RL and NE methods by tracking the elite_rate, selected_rate, and discard_rate of the transferred RL agents within the NE population. Table 1 shows the averaged selection rate during the training process for each baseline opponent. A high level of average discard_rate means that the transferred RL agents were mostly discarded and the NE population is still the mainstream of the NE_RL. The contributions of transferred RL agents are represented by elite_rate and selected_rate. Though it was only a small fraction, it indeed made a big difference for NE_RL, which outperformed both the NE and RL methods.



(**a**)

(**b**)

(**c**)

(**d**)

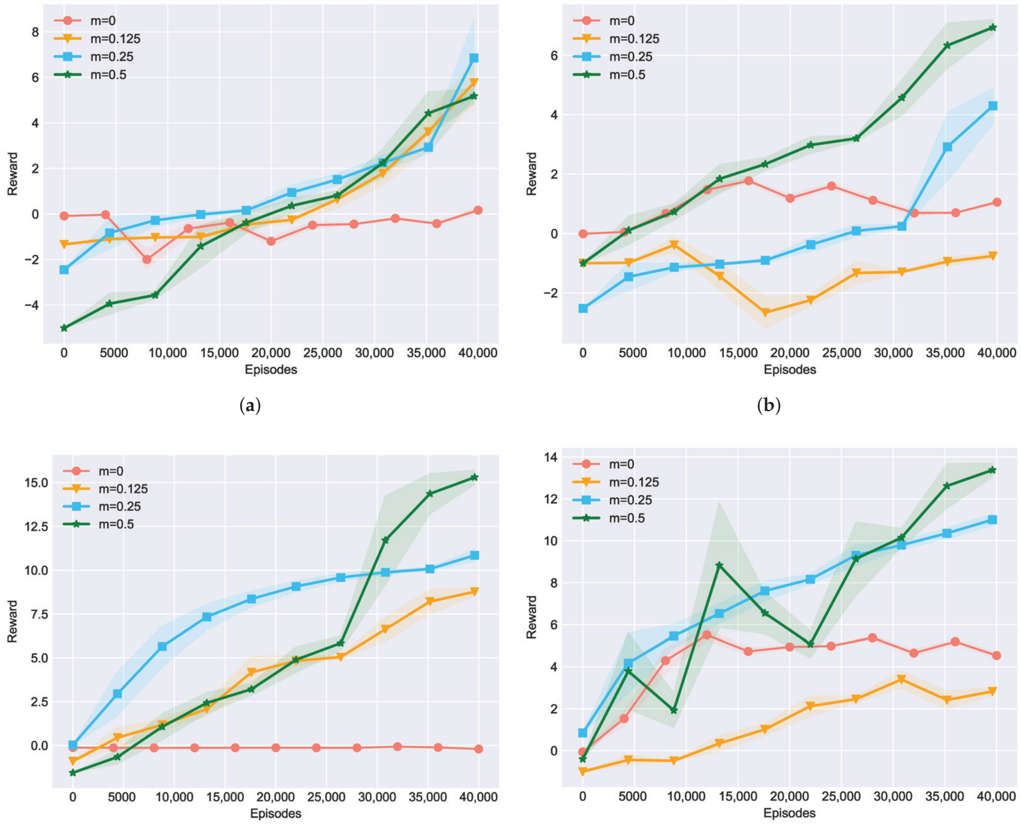
**Figure 6.** Learning curves of different methods in training against the four baseline opponents: (**a**) TA, (**b**) TP, (**c**) LA, (**d**) LP.

**Table 1.** Selection rate for transferred RL agents within the NE population.

| | **Elite** | **Selected** | **Discarded** |
|---|---|---|---|
| TA | 27.8 ± 2.7% | 28.1 ± 3.3% | 71.9 ± 2.5% |
| TP | 4.0 ± 2.8% | 5.7 ± 3.4% | 94.3 ± 3.2% |
| LA | 13.2 ± 6.0% | 26.2 ± 6.9% | 73.8 ± 10.3% |
| LP | 14.9 ± 8.7% | 24.4 ± 14.1% | 75.6 ± 13.3% |

### *4.3. Ablation Experiments*

Next, we used an ablation experiment to test how the strength of the transfer process affects NE_RL's performance. During the transfer process, the RL agents reinsert into the NE population to provide learned gradient information. We define the strength of this process as $m$, which represents the number of RL agents relative to the size of the NE population. Suppose the size of the NE population is fixed; then, a higher $m$ means a stronger impact imposed on the NE population by the RL agents. Here we represented $m$ as a fraction of the size of the NE population and conducted contrast experiments with different $m$ values. Figure 7 shows the experimental results in training against all four baseline opponents, with $m$ ranging from 0 to 0.5, which means the the strength of the transfer process changes from weak to strong. Typically, $m = 0$ represents the original NE method and $m = 0.5$ represents a rather strong NE_RL method, which was commonly used in the previous experiments. From those results, we can conclude that in general, different $m$ values may lead to visible performance differences, which indicates that the value of $m$ may be one of the most important hyper-parameters for NE_RL. In this experiment, it seems that a higher value of $m$ achieved better converging performance after a limited number of 40,000 training episodes. However, this may not be true in all cases. It depends on the specific experimental settings and we cannot draw a definite conclusion in this paper. It may be a good topic to study further in future work.



(**a**)

(**b**)

(**c**)

(**d**)

**Figure 7.** Ablation experiments on different RL population sizes, $m$, in training against the four baseline opponents: (**a**) TA, (**b**) TP, (**c**) LA, (**d**) LP.

## 5. Discussion

In this paper, we proposed a novel method for opponent exploitation in imperfect information games such as NLTH. Since a NLTH-like game typically contains challenges of sparse rewards and high complexity, the previous RL- or NE-only methods for opponent exploitation struggle with poor optimal performance or low sample efficiency. Our NE_RL method uses NE's advantage of evolutionary computation with a long-term fitness metric to address the sparse rewards feedback in NLTH and retains RL's gradient-based method for higher learning efficiency. Additionally, NE_RL recycles data generated by both NE and RL populations and uses an experience replay mechanism for the off-policy RL method to learn from them more than once, which can greatly help improve sample efficiency. Experimental results against various opponents show that NE_RL outperforms the previous NE- and RL-only methods with advantages of maximum exploitation and sample-efficient learning.

Apart from NLTH, we believe that our method can also be extended to other domain problems with challenges of adversarial sequential decision-making processes and imperfect information. Future work based on this paper includes incorporating more complex evolutionary sub-mechanisms to improve the standard NE operators used in NE_RL, or incorporating more advanced off-policy reinforcement learning methods and techniques to replace the currently used DQN in NE_RL. In addition, extending NE_RL to multiplayer NLTH will be another exciting thread of research leading to wider application prospects.

**Author Contributions:** Conceptualization, J.X.; methodology, J.X. and S.C.; validation, J.X. and S.C.; resources, J.C.; data curation, J.X.; writing—original draft preparation, J.X.; writing—review and editing, S.C.; supervision, J.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Our code will be available at https://github.com/jiahui-x/NE_RL (accessed on 14 July 2021).

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

## References

1. Brown, N.; Sandholm, T. Superhuman AI for multiplayer poker. *Science* **2019**, *365*, eaay2400. [CrossRef] [PubMed]
2. Bowling, M.; Burch, N.; Johanson, M.; Tammelin, O. Heads-up limit hold'em poker is solved. *Science* **2015**, *347*, 145–149. [CrossRef] [PubMed]
3. Sandholm, T. The state of solving large incomplete-information games, and application to poker. *Ai Mag.* **2010**, *31*, 13–32. [CrossRef]
4. Zha, D.; Lai, K.H.; Huang, S.; Cao, Y.; Hu, X. RLCard: A Platform for Reinforcement Learning in Card Games. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence IJCAI-PRICAI-20, Yokohama, Japan, 11–17 July 2020.
5. Brown, N.; Sandholm, T. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* **2018**, *359*, 418–424. [CrossRef] [PubMed]
6. Moravík, M.; Schmid, M.; Burch, N.; Lis, V.; Bowling, M. DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker. *Science* **2017**, *356*, 508. [CrossRef] [PubMed]
7. Maîtrepierre, R.; Mary, J.; Munos, R. Adaptive play in texas hold'em poker. In *ECAI 2008: 18th European Conference on Artificial Intelligence, Patras, Greece, 21–25 July 2008: Including Prestigious Applications of Intelligent Systems (PAIS 2008)*; IOS Press: Amsterdam, The Netherlands, 2008; Volume 178, p. 458.
8. Southey, F.; Bowling, M.P.; Larson, B.; Piccione, C.; Burch, N.; Billings, D.; Rayner, C. Bayes' bluff: Opponent modelling in poker. *arXiv* **2012**, arXiv:1207.1411.
9. Pricope, T.V. A View on Deep Reinforcement Learning in Imperfect Information Games. *Stud. Univ. Babeș-Bolyai Inform.* **2020**, *65*, 31. [CrossRef]
10. Brown, N.; Sandholm, T. Safe and nested endgame solving for imperfect-information games. In Proceedings of the Workshops at the thirty-first AAAI conference on artificial intelligence, San Francisco, CA, USA, 4–5 February 2017.
11. Li, X.; Miikkulainen, R. Evolving adaptive poker players for effective opponent exploitation. In Proceedings of the AAAI Workshops, San Francisco, CA, USA, 4–9 February 2017.
12. Nicolai, G.; Hilderman, R.J. No-Limit Texas Hold'em Poker agents created with evolutionary neural networks. In Proceedings of the International Conference on Computational Intelligence & Games, Milan, Italy, 7–10 September 2009.

13. Li, H.; Wang, X.; Jia, F.; Li, Y.; Chen, Q. A Survey of Nash Equilibrium Strategy Solving Based on CFR. *Arch. Comput. Methods Eng.* **2020**, *28*, 2749–2760. [CrossRef]
14. Lu, S. Online Enhancement of Existing Nash Equilibrium Poker Agents. Master's Thesis, Knowledge Engineering Group, Darmstadt, Germany, 2016.
15. Ganzfried, S.; Sandholm, T. Game theory-based opponent modeling in large imperfect-information games. In Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, Taipei, Taiwan, 2–6 May 2011; Volume 2, pp. 533–540.
16. Gilpin, A.; Sandholm, T. *Finding Equilibria in Large Extensive form Games of Imperfect Information*; Technical Report; Mimeo: New York, NY, USA, 2005.
17. Ganzfried, S.; Sandholm, T. Safe Opponent Exploitation. *ACM Trans. Econ. Comput.* **2012**, *3*, 587–604.
18. Teófilo, L.; Reis, L.P. Identifying Player's Strategies in No Limit Texas Hold'em Poker through the Analysis of Individual Moves. *arXiv* **2013**, arXiv:1301.5943.
19. Bard, N.; Johanson, M.; Burch, N.; Bowling, M. Online implicit agent modelling. In Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems, St. Paul, MN, USA, 6–10 May 2013; pp. 255–262.
20. Ackley, D. Interactions between learning and evolution. *Artif. Life II* **1991**, *11*, 487–509
21. Munir-ul, M.C.; Yun, L. Evolutionary reinforcement learning for neurofuzzy control. In Proceedings of the International Fuzzy Systems Association World Congress, Prague, Czech Republic, 25–29 June 1997
22. Lin, C.J.; Hsu, Y.C. Reinforcement hybrid evolutionary learning for recurrent wavelet-based neurofuzzy systems. *IEEE Trans. Fuzzy Syst.* **2007**, *15*, 729–745. [CrossRef]
23. Koppejan, R.; Whiteson, S. Neuroevolutionary reinforcement learning for generalized control of simulated helicopters. *Evol. Intell.* **2011**, *4*, 219–241. [CrossRef]
24. Drugan, M.M. Reinforcement learning versus evolutionary computation: A survey on hybrid algorithms. *Swarm Evol. Comput.* **2018**, *44*, 228–246. [CrossRef]
25. Khadka, S.; Tumer, K. Evolution-Guided Policy Gradient in Reinforcement Learning. *arXiv* **2018**, arXiv:1805.07917.
26. Floreano, D.; Dürr, P.; Mattiussi, C. Neuroevolution: From architectures to learning. *Evol. Intell.* **2008**, *1*, 47–62. [CrossRef]
27. Salimans, T.; Ho, J.; Chen, X.; Sidor, S.; Sutskever, I. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. *arXiv* **2017**, arXiv:1703.03864.
28. Ketkar, N. Introduction to PyTorch. In *Deep Learning with Python*; Apress: Berkeley, CA, USA, 2017.
29. Lisy, V.; Bowling, M. Equilibrium Approximation Quality of Current No-Limit Poker Bots. *arXiv* **2016**, arXiv:1612.07547.
30. Li, K.; Xu, H.; Zhang, M.; Zhao, E.; Wu, Z.; Xing, J.; Huang, K. OpenHoldem: An Open Toolkit for Large-Scale Imperfect-Information Game Research. *arXiv* **2020**, arXiv:2012.06168.

*electronics*

MDPI

*Article*

# Evolutionary Multiobjective Optimization with Endmember Priori Strategy for Large-Scale Hyperspectral Sparse Unmixing

**Zhao Wang [1,*], Jinxin Wei [1], Jianzhao Li [1], Peng Li [1] and Fei Xie [2]**

[1] Key Laboratory of Electronic Information Countermeasure and Simulation Technology of Ministry of Education, Xidian University, No. 2 South TaiBai Road, Xi'an 710071, China; jinxinwei@stu.xidian.edu.cn (J.W.); 19jzli@stu.xidian.edu.cn (J.L.); li_peng001@163.com (P.L.)

[2] Academy of Advanced Interdisciplinary Research, Xidian University, No. 2 South TaiBai Road, Xi'an 710071, China; fxie@xidian.edu.cn

[*] Correspondence: wangzhao@xidian.edu.cn; Tel.: +86-186-2930-7629

**Abstract:** Mixed pixels inevitably appear in the hyperspectral image due to the low resolution of the sensor and the mixing of ground objects. Sparse unmixing, as an emerging method to solve the problem of mixed pixels, has received extensive attention in recent years due to its robustness and high efficiency. In theory, sparse unmixing is essentially a multiobjective optimization problem. The sparse endmember term and the reconstruction error term can be regarded as two objectives to optimize simultaneously, and a series of nondominated solutions can be obtained as the final solution. However, the large-scale spectral library poses a challenge due to the high-dimensional number of spectra, it is difficult to accurately extract a few active endmembers and estimate their corresponding abundance from hundreds of spectral features. In order to solve this problem, we propose an evolutionary multiobjective hyperspectral sparse unmixing algorithm with endmember priori strategy (EMSU-EP) to solve the large-scale sparse unmixing problem. The single endmember in the spectral library is used to reconstruct the hyperspectral image, respectively, and the corresponding score of each endmember can be obtained. Then the endmember scores are used as a prior knowledge to guide the generation of the initial population and the new offspring. Finally, a series of nondominated solutions are obtained by the nondominated sorting and the crowding distances calculation. Experiments on two benchmark large-scale simulated data to demonstrate the effectiveness of the proposed algorithm.

**Keywords:** large-scale multiobjective optimization; sparse unmixing; hyperspectral image; evolutionary algorithm

**Citation:** Wang, Z.; Wei, J.; Li, J.; Li, P.; Xie, F. Evolutionary Multiobjective Optimization with Endmember Priori Strategy for Large-Scale Hyperspectral Sparse Unmixing. *Electronics* **2021**, *10*, 2079. https://doi.org/10.3390/electronics10172079

Academic Editor: George A. Papakostas

Received: 17 July 2021
Accepted: 24 August 2021
Published: 27 August 2021

## 1. Introduction

Hyperspectral imagery, which contains a wealth of spectral information for the surface features in each pixel, has been widely used in various remote sensing applications, such as geological analysis, environmental monitoring and military reconnaissance. However, due to the low spatial resolution and the ground substances intimate mixtures, the mixed pixels inevitably appear in the hyperspectral images. To solve this problem, the spectral unmixing technique aims to extract the pure spectral signatures (also called endmembers) from hyperspectral images and estimate their corresponding proportions (also called abundances).

The spectral unmixing assumes that there is no multiple scattering between endmembers in the spectrum, each pixel is a linear combination of elements from the endmember set in the linear mixed model (LMM) [1]. Under this model, various methods such as geometry-based [2], nonnegative matrix factorization-based (NMF) [3–5] and statistical-based [6] have been conducted research in the hyperspectral spectral unmixing, which also obtained a very ideal unmixing effect. However, these methods suffer from poor performance when the assumption of pure pixels or the generation of virtual endmembers do not satisfy.

As an emerging spectral unmixing technology, Hyperspectral sparse unmixing aims to find the optimal subset of the true endmembers for reconstructing the mixed pixels based on the known spectral library in advance. Compared to the number of endmembers in the spectral library, the number of endmembers use for reconstructing is is relatively sparse. Mathematically speaking, it is a $l_0$ norm problem, which is highly non-convex and NP-hard [7]. The relaxation methods, such as the $l_1$ norm or the $l_p$ norm ($0 < p < 1$), are some of the relatively effective solutions to deal with the $l_0$ norm problem. Bioucas-Dias et al. [8] solves the sparse unmixing problem by the alternating direction method of multipliers, but the SUnSAL [8] only focuses on the spectral information without taking the spatial structure information between different pixels into account. To take advantage of the relationship between pixels in the hyperspectral image, many strategies such as collaborative sparse regression framework [9] or spectral regularization terms [10,11] are applied in the sparse unmixing model to promote the spatial correlation.

However, these algorithms are very sensitive to parameter settings, which greatly affect the stability of unmixing algorithms. Recently, the intelligent optimization algorithms have been greatly developed. Połap et al. [12] proposed a red fox optimization algorithm by simulating the hunting behavior of fox. In [13], a polar bear optimization algorithm was proposed to simulate the hunting behavior of a polar bear into the stage of global search and the stage of local search. Khishe et al. [14] proposed a chimp optimization algorithm to further alleviate the two problems of slow convergence speed and trapping in local optima in solving high-dimensional problems. Nevertheless, in solving the NP hard problems, the multiobjective evolutionary algorithms (MOEAs) have attracted extensive attention because of the global search ability. Therefore, some excellent algorithms proposed in [15–18] have applied the multiobjective optimization to sparse unmixing. However, most of MOEA-based algorithms in sparse unmixing only focus on solving pixel by pixel efficiently without considering the spatial structure information because of the curse of dimensions. In addition, MOEA-based algorithms suffer from the problem of being time-consuming, resulting in inefficiency and impracticality. Compared with the large-scale spectral library, the number of endmembers used to reconstruct the hyperspectral image is actually sparse. Therefore, many existing MOEAs suffer from a large number of decision variables when dealing with the sparse multiobjective optimization problems, which consumes the expensive computing resources to search in a large decision space with an arbitrary initialization.

To alleviate the above problems, an evolutionary multiobjective hyperspectral sparse unmixing algorithm with endmember priori strategy (EMSU-EP) is proposed in this paper. In the proposed EMSU-EP, each decision variable is taken out separately for evaluation in the initialization, and their corresponding scores are recorded for the subsequent crossover and mutation. In the subsequent genetic operation, a new genetic operator is designed to ensure the sparsity of the offspring in a uniform interval. With the prior knowledge of the quality of each decision variable and the new genetic operators for binary variables, the proposed EMSU-EP can achieve the better convergence performance and diversity, and the result of unmixing has been greatly improved. The main contributions of the proposed EMSU-EP are summarized as follows.

(1) We propose a novel multiobjective optimization framework for sparse unmixing, which can guide the subsequent evolution of the algorithm according to the prior knowledge obtained from the spectrum library.

(2) A special initialization mechanism is designed, it is demonstrated that the proposed EMSU-EP can obtain the diverse and targeted population compare with the state-of-the-art MOEA-based sparse unmixing methods.

(3) The particular crossover and mutation operators are proposed to maintain the sparsity of the population, which can not only promote the convergence of the algorithm, but also improve the performance of sparse unmixing.

The rest of this paper is organized as follows. In Section 2, the related works are summarized. In Section 3, the framework of our proposed EMSU-EP algorithm is introduced in detail. The experimental results are presented and analyzed in Section 4. Finally, the work in this paper is concluded in Section 5.

## 2. Related Works

In this section, the related works on sparse unmixing and the multiobjective optimization are introduced.

### *2.1. Sparse Unmixing*

As shown in Figure 1, sparse unmixing aims to find the optimal set of endmembers for modeling the mixed pixels from a large-scale and pre-known spectral library. Therefore, a mixed pixel ($\boldsymbol{y} \in \mathbb{R}^{L\times 1}$) with $L$ spectral bands can be formulated as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{L\times D}$ is the spectral library, $\boldsymbol{x} \in \mathbb{R}^{D\times 1}$ represents the corresponding fractional abundance vector and $\boldsymbol{n} \in \mathbb{R}^{L\times 1}$ is the noise term. In the absence of noise, the sparse unmixing method of the Formula (1) is mathematically expressed as

$$\begin{aligned} &\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \\ s.t.\ \ &\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 \le \delta \end{aligned} \tag{2}$$

where $\delta \ge 0$ denotes the error tolerance, $\|\boldsymbol{x}\|_0$ is the $l_0$ norms of $\boldsymbol{x}$, which is highly non-convex and NP-hard. It was not well solved until Candes et al. [7] proved that $l_1$ norm can induce sparsity instead of $l_0$ norm under a certain restricted isometry property condition. In [8], the SUnSAL was proposed to solve the sparse unmixing problem of mixed pixels by establishing constrained sparse regression as

$$\min_{\boldsymbol{x}} (1/2)\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 + \iota_{R+}(\boldsymbol{x}) + \iota_{\{1\}}(\mathbf{1}^{\mathbf{T}}\boldsymbol{x}) \tag{3}$$

where $\|\boldsymbol{x}\|_1$ denotes the $l_1$ norm of $\boldsymbol{x}$, $\lambda$ is a regularization parameter that controls the relative weight of the error term and the sparse term, $\mathbf{1}$ denotes a column vector of 1's, the $\iota_{R+}(\boldsymbol{x})$ and $\iota_{\{1\}}(\mathbf{1}^{\mathbf{T}}\boldsymbol{x})$ represent the abundance non-negativity constraint (ANC) and the abundance sum-to-one constraint (ASC) [19], respectively.

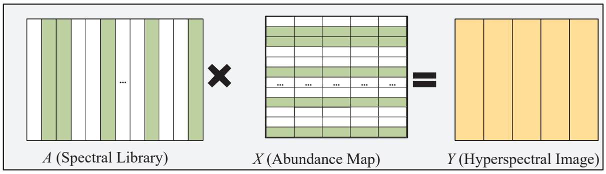


**Figure 1.** Illustration of the hyperspectral sparse unmixing problem, $Y$ is the image to be unmixed, $A$ is the large-scale spectral library, and $X$ is the obtained abundance matrix. Only colored active endmembers participate in the reconstruction of endmembers.

However, the SUnSAL [8] only focuses on the spectral information without taking the spatial structure information between different pixels into account. In general, a hyperspectral image ($\boldsymbol{Y} \in \mathbb{R}^{L\times n}$) with $n$ pixels structured in the matrix can be formulated as

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{N} \tag{4}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n]$, $\boldsymbol{y}_i$ is the $i$-th mixed pixel, $\boldsymbol{X} \in \mathbb{R}^{D \times n}$ is the abundance matrix, and $\boldsymbol{N} \in \mathbb{R}^{L \times n}$ is the corresponding error term. The formula (4) can be transformed into an optimization problem expressed as

$$\begin{aligned} &\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_0 \\ &s.t. \quad \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|_F^2 \leq \delta \end{aligned} \tag{5}$$

where $\|\boldsymbol{X}\|_0 = |\text{supp}(\boldsymbol{X})|$, $\text{supp}(\boldsymbol{X}) = \{1 \leq i \leq D, \mathbf{x}_i \neq \mathbf{0}\}$, $\boldsymbol{x}_i$ is $i$th row of $\boldsymbol{X}$. To take advantage of the relationship between pixels in the hyperspectral image, CLSUnSAL [9] assumes that all pixels share the same active endmembers to reduce the influence of the spectral coherence between endmembers on the unmixing effect. The model is shown as follows

$$\min_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|_F^2 + \lambda \|\boldsymbol{X}\|_{2,1} + \iota_{R+}(\boldsymbol{X}) \tag{6}$$

where $\|\mathbf{X}\|_{2,1} = \sum_{k=1}^{m} \left\|\mathbf{x}^k\right\|_2$ represents the $l_{2,1}$ norm, $\iota_{R+}(X) = \sum_{i=1}^{n} \iota_{R+}(\mathbf{x}_i)$ denotes the indicator function. Moreover, many spectral regularization terms, such as the total variation regularization term [10] and the non-local regularization terms [11], are integrated into the sparse unmixing model to promote the spatial correlation.

### 2.2. Multiobjective Optimization

The multiobjective evolutionary algorithms (MOEAs) have attracted extensive attention because of the global search ability in solving the sparse unmixing problems. In [15], Gong et al. proposed the multiobjective sparse unmixing model with a cooperative coevolutionary strategy. Jiang et al. [16] formulated the sparse unmixing problem into a two-phase multiobjective problem to estimate the endmembers and determine the abundances, respectively. In [17,18], the multiobjective evolutionary algorithm based on decomposition (MOEA/D) [20] has also been explored and applied in the sparse unmixing problem. For the multiobjective optimization problem, the mathematical form with $a$ objectives and $b$ decision variables can be described as

$$\begin{aligned} &\min F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_a(\boldsymbol{x}))^T \\ &s.t.\ \boldsymbol{x} = (x_1, x_2, \ldots, x_b)^T \in \boldsymbol{\Omega} \end{aligned} \tag{7}$$

where $\boldsymbol{x}$ is the decision vector, $F$: $\boldsymbol{\Omega} \rightarrow \mathbb{R}^a$, $\boldsymbol{\Omega}$ is the decision space and $\mathbb{R}^a$ is the objective space.

In the majority of cases, there is no single solution capable of minimizing all the objectives at the same time. Instead, the best trade-off between the objectives can be defined as Pareto optimality. Therefore, in order to evaluate the pros and cons of multiple solutions, the nondominated fronts and crowding distances of the individual can be applied. If the individuals are dominated by the same number of individuals, these individuals belong to the same nondominated front. In addition, the crowding distance of an individual can be obtained by calculating the side length of the rectangle formed between two adjacent individuals that belong to the same nondominated front with the individual. Assuming that $p_1$ and $p_2$ are two individuals in the population, individual $p_2$ is preferred over $p_1$ (i.e., $p_2 \succ p_1$) if any one of the following conditions holds [21], (1) $\text{NF}_2 < \text{NF}_1$, (2) $\text{NF}_2 = \text{NF}_1$ and $\text{CD}_2 > \text{CD}_1$, where $\text{NF}_1$ and $\text{NF}_2$ represent the nondominated fronts of individuals $p_1$ and $p_2$, respectively. $\text{CD}_1$ and $\text{CD}_2$ represent the crowding distances of individuals $p_1$ and $p_2$, respectively.

## 3. Proposed Method

For the sparse unmixing problem, there are two conflicting objectives to be optimized, namely the sparse endmember term and the reconstruction error term. Therefore, the multiobjective optimization problem for sparse unmixing is

$$\min_{\boldsymbol{X}} (\|\boldsymbol{X}\|_0, \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|_F^2) \tag{8}$$

where the $\|\boldsymbol{X}\|_0$ is the sparse term and the $\|\boldsymbol{Y} - \boldsymbol{AX}\|_F^2$ represents the reconstruction error. The Formula (8) can be solved with the multiobjective optimization to obtain the Pareto Front (PF) of a compromise between these two objectives. Then the knee point is selected as the final solution, which is the preferred solution on PF with the maximum marginal utility and can be obtained from the individual with the maximum angle with the two adjacent individuals. In addition, two constrains for the abundance are required by

$$\begin{cases} \text{ANC} : \boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \geq \boldsymbol{0} \\ \text{ASC} : \forall \boldsymbol{x}_i \in \boldsymbol{X}, \boldsymbol{1}^T \boldsymbol{x}_i = 1, i = 1, \ldots, n \end{cases} \tag{9}$$

The pseudocode of the proposed EMSU-EP is shown in Algorithm 1. First of all, in the initialization, the $k$-th endmember is extracted from the spectral library at a time to reconstruct the hyperspectral image $\hat{\boldsymbol{Y}}_k$ for obtaining the corresponding reconstruction error with the original image. This operation needs to run through all the endmembers in the spectral library. The reconstruction errors of all endmembers are sorted in ascending order, and the corresponding $Score_k$ represents the position order. Then the spectral library is encoded into a binary vector with dimension $D$, where "1" and "0" represent the selected and unselected endmembers, respectively. Then EMSU-EP randomly selects two elements from the D-dimensional decision variables each time, and uses the endmember score as the evaluation criterion to set one of to "1" with the binary tournament selection method, which is shown in Figure 2. In order to ensure the diversity of the initial population, the *rand()* operator is employed to uniformly distribute $N$ individuals in the sparse interval [0, $D$], respectively. Unlike the inefficiency of random initialization, the more likely real endmembers can be selected to form the initial population with the prior knowledge.

**Algorithm 1** Pseudocode of EMSU-EP

```
Input: $A$ (the spectral library), $Y$ (the hyperspectral image), $N$ (population size), $t_{max}$ (maximum number of iterations).
Output: $X^*$ (the estimated abundance map).
for $k = 1 : D$ do
    Reconstruct $\hat{Y}_k$ with the $k$-th endmember in $A$.
    $Score_k \leftarrow \|Y - \hat{Y}_k\|_F^2$
end for
Encode $A$ as a binary vector;
% $P \leftarrow Initialization(N)$
for $i = 1 : N$ do
    for $j = 1 : rand() \times D$ do
        $[m, n] \leftarrow$ Randomly select two endmembers;
        if $Score_m < Score_n$ then
            Set the $m$-th endmember to 1 in $i$-th individual;
        else
            Set the $n$-th endmember to 1 in $i$-th individual;
        end if
    end for
end for
$t \leftarrow 0$;
while $t \leq t_{max}$ do
    $C \leftarrow Mutation(Crossover(P, N))$;
    $P \cup C \leftarrow Evaluation(P \cup C, 2N)$;
    $P \leftarrow Selection(P \cup C, N)$;
    $t \leftarrow t + 1$;
end while
```

With the prior knowledge of decision variables, those decision variables with higher scores should be given more attention in the subsequent genetic operations. Therefore, the genetic operators of crossover and mutation proposed in [22] are employed to generate the offspring, which are represented by Crossover() and Mutation(), respectively, in Algorithm 1. The left gray box and the the right gray box represent the Crossover() operator and the Mutation() operator, respectively, in Figure 3. Before evolution, two individuals $p_1$ and $p_2$ are randomly selected from population as the parents. There are two situations in the Crossover(). On the one hand, the original offspring $c$ inherits $p_1$, then two non-zero elements are randomly selected from the differential gene positions of $p_1$ and $p_2$, and one of the gene positions of $c$ is set to "0" based on endmember scores (the larger the better). On the other hand, two zero elements are randomly selected from the differential gene positions of $p_1$ and $p_2$, and one of the gene positions of $c$ is set to "1" based on endmember scores (the smaller the better). In the Mutation(), two non-zero elements are randomly selected from the offspring $c$, and one of the gene positions is set to "0" determined by the larger score. To the opposite, two zero elements are randomly selected from $c$, and set one of the gene positions is set to "1" determined by the smaller score. A zero or non-zero element in the binary vector is flipped with the same probability as shown in Figure 3. Compared with the single-point crossover (SPC) and bitwise mutation (BitM), the operators designed by the EMSU-EP select the element to be flipped according to the score of the decision variable to ensure the sparsity of the offspring.

During the evolution of the population, if too many endmembers are selected, some solutions are no longer sparse. To prevent this problem from happening, the sparsity limit $d$ is applied to the population evolution. When the number of selected endmembers in an individual exceeds the sparsity limit $d$, EMSU-EP will only retain the endmember whose endmember score sorting in the first $d$-th, and the rest will not be selected (i.e., set to 0).

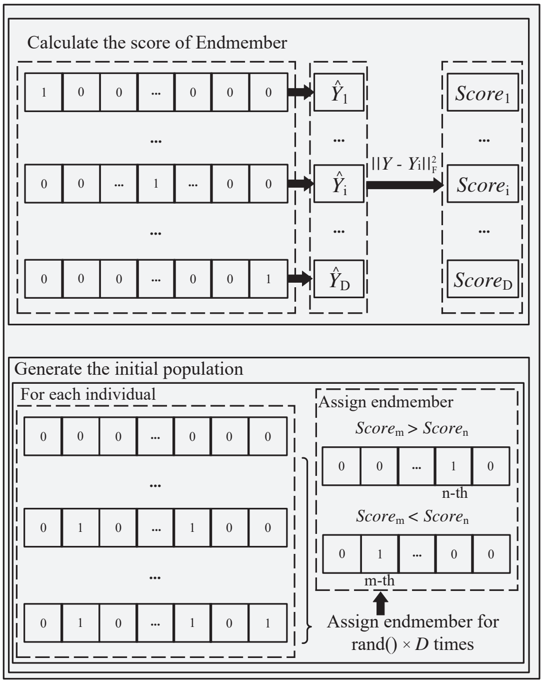


**Figure 2.** Obtaining the score of every endmember and generate the initial population.

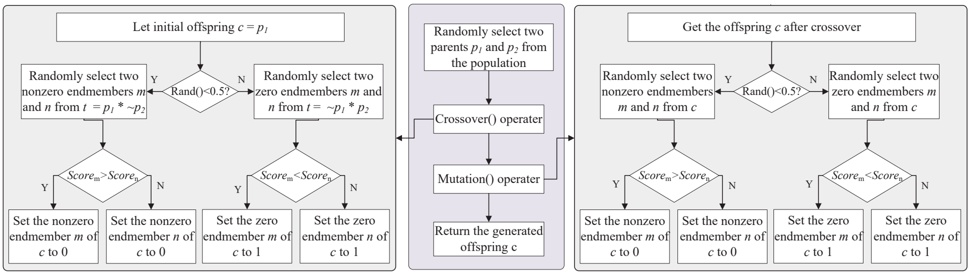

Let initial offspring $c = p_1$
Randomly select two nonzero endmembers $m$ and $n$ from $t = p_1 * \sim p_2$
Rand()<0.5?
Y
N
Randomly select two zero endmembers $m$ and $n$ from $t = \sim p_1 * p_2$
$Score_m > Score_n$
Y
N
$Score_m < Score_n$
Y
N
Set the nonzero endmember $m$ of $c$ to 0
Set the nonzero endmember $n$ of $c$ to 0
Set the zero endmember $m$ of $c$ to 1
Set the zero endmember $n$ of $c$ to 1
Randomly select two parents $p_1$ and $p_2$ from the population
Crossover() operater
Mutation() operater
Return the generated offspring c
Get the offspring $c$ after crossover
Randomly select two nonzero endmembers $m$ and $n$ from $c$
Y
Rand()<0.5?
N
Randomly select two zero endmembers $m$ and $n$ from $c$
$Score_m > Score_n$
Y
N
$Score_m < Score_n$
Y
N
Set the nonzero endmember $m$ of $c$ to 0
Set the nonzero endmember $n$ of $c$ to 0
Set the zero endmember $m$ of $c$ to 1
Set the zero endmember $n$ of $c$ to 1

**Figure 3.** Illustration of the flow of crossover and mutation operators. The middle procedure represents the generation process of a offspring, the left procedure represents the crossover operation, and the right procedure represents the mutation operation.

Finally, all individuals of the parent and offspring are evaluated for nondominated sorting and crowding distance calculation [23], and the best $N$ individuals are selected to form the next generation population. After satisfying the end of the iteration, the knee point in the last generation population is returned [24]. The set of non-zero elements are extracted from this optimal binary vector, which also represents the corresponding endmember subset $\boldsymbol{A}_{S^*}$ from the spectral library $\boldsymbol{A}$. Therefore, the abundance map $\boldsymbol{X}_{S^*}$ can be calculated according to the least square method, which is shown as follows

$$\boldsymbol{X}_{S^*} = \arg\min_{\boldsymbol{X}_{S^*}} \|\boldsymbol{Y} - \boldsymbol{A}_{S^*}\boldsymbol{X}_{S^*}\|_F. \tag{10}$$

According to the Formula (10), we can not only achieve dimensionality reduction of hyperspectral data, but also ensure the sparseness of unmixing solutions. After the calculation of Formula (10), the zero element is inserted into the non-zero real solution according to the original position to realize the restoration of the data dimension.

## 4. Experimental Results and Discussion

In this section, the effectiveness of EMSU-EP in solving large-scale sparse unmixing problems will be demonstrated. Two large-scale sparse unmixing benchmark datasets are used to validate the performance of the EMSU-EP. In order to reflect the efficiency of EMSU-EP, some advanced algorithms such as SUnSAL [8], CLSUnSAL [9], MOSU [15] and MTSR [21] will be compared with EMSU-EP.

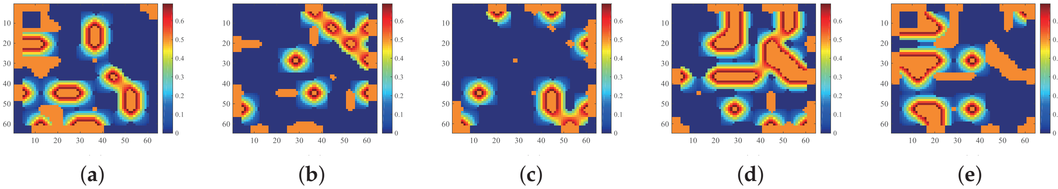
In the experiment, the population size is set to 100, and the maximum generation Maxgen is set to 300, the population sparse limit interval $d$ is 50. Part of the experiment code refers to the PlatEMO platform (PlatEMO 2.8: https://github.com/BIMK/PlatEMO (accessed on 22 August 2021)) [25]. All experiments will be added with different degrees of Gaussian white noise (SNR = 20, 30 and 40 dB). All experimental results are obtained from the average of 100 independent repeated runs.

### *4.1. Dataset and Evaluation Indicators*

#### 4.1.1. Dataset

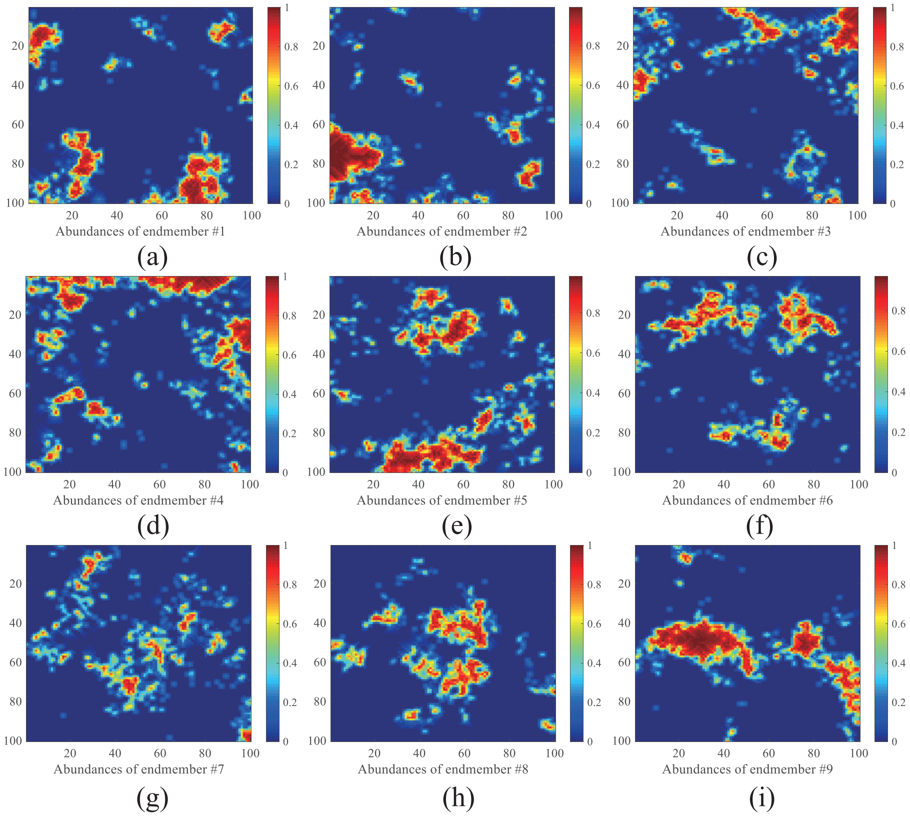
Data 1 is a 64 × 64 synthetic image containing 224 bands provided by Tang [26], its digital spectral library A1 is a sub-library of 498 spectral features selected from the USGS spectral library (http://speclab.cr.usgs.gov/spectral.lib06 (accessed on 22 August 2021)). These spectral signals are evenly distributed at 0.25–0.4 µm. The true abundance map of all five endmembers of data 1 is shown in Figure 4. Data 2 is an image of 100 × 100 pixels and 224 bands per pixel, provided by Iordache et al. [10], its digital spectral library A2 is a sublibrary with 230 spectral signatures of the USGS spectral library. The true abundance map of all nine endmembers of data 2 is shown in Figure 5.

To summarize, data 1 needs to accurately select five endmembers from 498 spectral signals and estimate the corresponding abundance, and data 2 needs to select nine endmembers from 230 spectral signals and estimate the corresponding abundance. Two datasets are sparse enough and difficult. Therefore, both data 1 and data 2 are large-scale sparse unmixing problems.



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

**Figure 4.** True abundance maps of five endmembers in data 1. (**a**) True abundance map of endmember 1. (**b**) True abundance map of endmember 2. (**c**) True abundance map of endmember 3. (**d**) True abundance map of endmember 4. (**e**) True abundance map of endmember 5.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

**Figure 5.** True abundance maps of five endmembers in data 2. (**a**) True abundance map of endmember 1. (**b**) True abundance map of endmember 2. (**c**) True abundance map of endmember 3. (**d**) True abundance map of endmember 4. (**e**) True abundance map of endmember 5. (**f**) True abundance map of endmember 6. (**g**) True abundance map of endmember 7. (**h**) True abundance map of endmember 8. (**i**) True abundance map of endmember 9.

#### 4.1.2. Evaluation Indicators

In order to compare the accuracy and the robustness of different sparse unmixing algorithms on large-scale hyperspectral sparse unmixing problem, two evaluation indicators are considered in the experiments.

(1) Signal-to-Reconstruction Error (SRE) can be expressed as:

$$\mathrm{SRE}\ (\mathrm{dB}) \equiv 10\log_{10}\left(\frac{E\left[\|\mathbf{X}\|_F^2\right]}{E\left[\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2\right]}\right) \tag{11}$$

where $\mathbf{x}$ is the true fractional abundance matrix and $\hat{\mathbf{x}}$ is the estimated fractional abundance matrix. Without loss of generality, the larger the SRE value is, the better the unmixing accuracy will be.

(2) Success Ratio (SR): If the relative error is smaller than a given threshold $\tau$, the corresponding run of this method is denoted as a successful run [27]. SR under the threshold $\tau$ is defined as:

$$\mathrm{SR}_\tau \equiv \mathrm{P}\left(\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2}{\|\mathbf{X}\|_F^2} \le \tau\right) \tag{12}$$

The probability is the ratio of the successful runs on 100 random instances. If we set $\tau = 5$ and arrive at $\mathrm{SR}_\tau = 1$, this implies that the relative error of the reconstruction result is less than 5 with probability one.

The Hypervolume (HV) [28] indicator can be used to evaluate the quality of PF, which can reflect the convergence and diversity of the solutions. HV is calculated by utilizing a reference point who is 1% larger in every component than the corresponding nadir point. In this paper, we use it to evaluate the performance on large-scale sparse unmixing problem by crossover and mutation operations of EMSU-EP, traditional SPC and BitM.

### *4.2. Experiments on Synthetic Data*

Figure 6 shows the PF of the 50-th generation population obtained by different methods on data 1 and data 2. Whether compared to the PF obtained by SPC and BitM with the initial population of EMSU-EP or the PF obtained by SPC and BitM with the random initial population, EMSU-EP obviously has better convergence and a clearer knee point area, which will be very helpful to choose the final solution later.

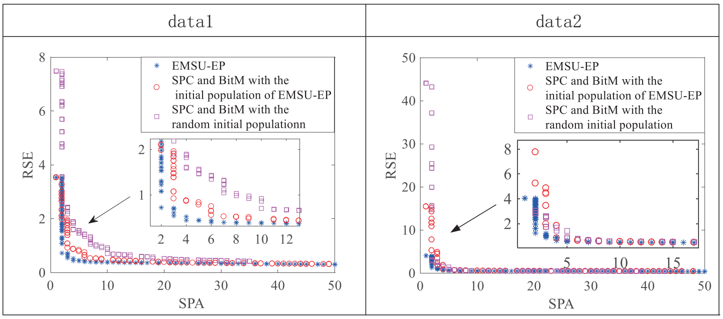


**Figure 6.** Illustration of the Pareto Front of the 50th generation on data 1 and data 2.

In order to reflect the advantages of EMSU-EP in each generation, Figure 7 uses the HV indicator to evaluate the PF of the first 100 generations. As shown by the red and purple curves in Figure 7, since the SPC and BitM operation has the same initial population of EMSU-EP, a higher HV value can be obtained from the first generation compared to the SPC and BitM with the random initial population, which corresponds to a better initial population quality and shows the guiding effect of endmember scores on the production of initial population. However, as shown by the blue and red curves in Figure 7, after the 20-th generation on data 1 and the 50-th generation on data 2, there are large static difference between the HV value of EMSU-EP and the HV value of the SPC and BitM operation, which shows the guiding effect of endmember scores on the evolution of the population.

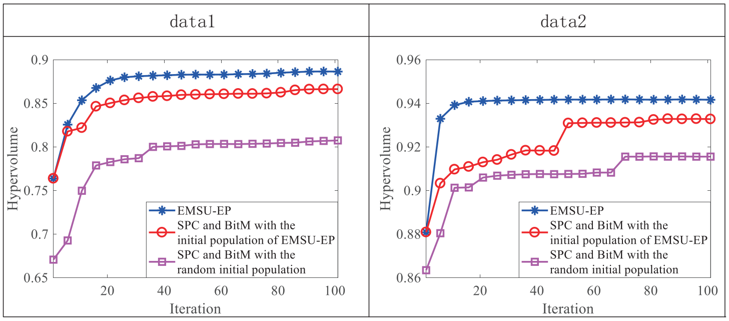


**Figure 7.** Illustration of the HV value of each evolution step on data 1 and data 2.

The estimated abundance maps of some endmembers obtained by different algorithms on data 1 and data 2 are shown in Figures 8 and 9, respectively. These experimental results are all on the 30dB SNR. In Figure 8, the abundance maps obtained by SUnSAL, CLSUnSAL, MOSU, MTSR and EMSU-EP are exhibited from left to right, endmember 1, endmember 2 and endmember 5 of data 1 are arranged from top to bottom. In Figure 9, the abundance maps obtained by SUnSAL, CLSUnSAL, MOSU, MTSR, and EMSU-EP are exhibited from left to right, endmember 1, endmember 5 and endmember 8 of data 2 are arranged from top to bottom. As shown in Figures 8 and 9, it is obvious to see that EMSU-EP always has the best performance compared to other algorithms, the unmixing maps color of EMSU-EP is the closest to the real image. Nevertheless, the performance of EMSU-EP in reducing noise is not stable enough, only the abundance map of endmember 1 has the least noise points in Figure 8, and only the abundance map of endmember 8 has the least noise points in Figure 9. Tables 1 and 2 show the SRE (dB) and $SR_\tau$ of the unmixing results obtained by different algorithms on data1 and data 2, respectively. The two data sets are corrupted by different levels of correlated noise (SNR = 20, 30 and 40 dB). According to Tables 1 and 2, EMSU-EP has the highest SRE (dB) and $SR_\tau$ compared to other algorithms at three SNR levels, which indicates that EMSU-EP has the best unmixing accuracy and robustness. The experimental results of two hyperspectral datasets demonstrate that the proposed EMSU-EP method can improve the performance of the sparse unmixing model by utilizing the endmember prior information.

**Table 1.** Comparison of EMSU-EP with other algorithms on data 1. The value of $\tau$ is set to 0.15.

| **Method** | **SRE (dB)** | | | **$SR_\tau$** | | |
|---|---|---|---|---|---|---|
| | **20** | **30** | **40** | **20** | **30** | **40** |
| SUnSAL | −4.3472 | 4.0704 | 13.8207 | 0.15 | 0.43 | 0.74 |
| CLSUnSAL | 8.2830 | 13.1349 | 14.3583 | 0.65 | 0.83 | 0.88 |
| MOSU | 7.3594 | 13.1434 | 14.4707 | 0.69 | 0.87 | 0.90 |
| MTSR | 10.7254 | 14.6143 | 17.6775 | 0.75 | 0.89 | 0.93 |
| EMSU-EP | **16.4900** | **19.9379** | **23.7559** | **0.78** | **0.91** | **0.95** |

**Table 2.** Comparison of EMSU-EP with other algorithms on data 2. The value of $\tau$ is set to 0.15.

| **Method** | **SRE (dB)** | | | **SR$_\tau$** | | |
|---|---|---|---|---|---|---|
| | **20** | **30** | **40** | **20** | **30** | **40** |
| SUnSAL | −4.2856 | 4.0604 | 13.3420 | 0.23 | 0.35 | 0.54 |
| CLSUnSAL | 5.5443 | 11.5608 | 18.9487 | 0.62 | 0.75 | 0.91 |
| MOSU | 5.5623 | 11.1713 | 19.4105 | 0.66 | 0.78 | 0.94 |
| MTSR | 7.0496 | 13.7802 | 22.7329 | 0.70 | 0.80 | 1.00 |
| EMSU-EP | **7.2801** | **14.6303** | **23.1275** | **0.74** | **0.86** | **1.00** |

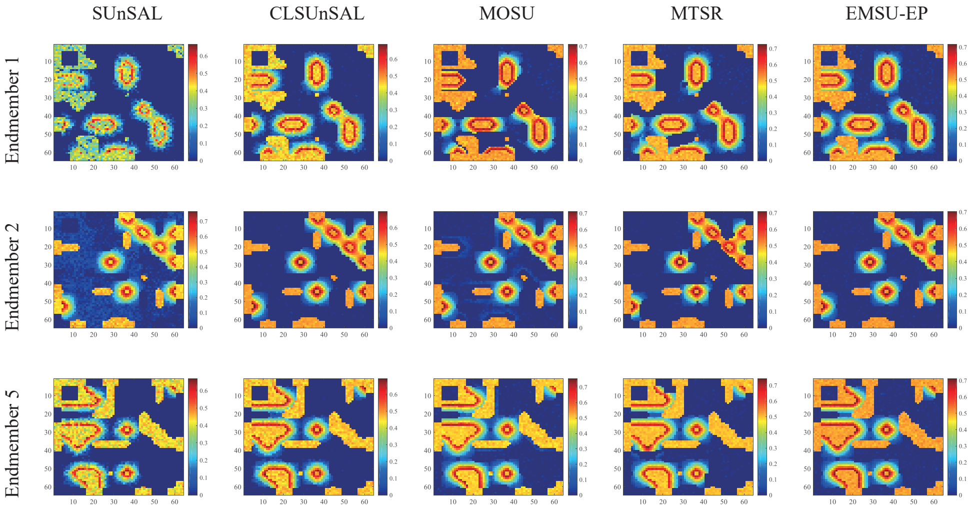


**Figure 8.** The estimated abundance maps of endmember 1, endmember 2 and endmember 5 for data 1 on 30 dB SNR obtained by different algorithms.

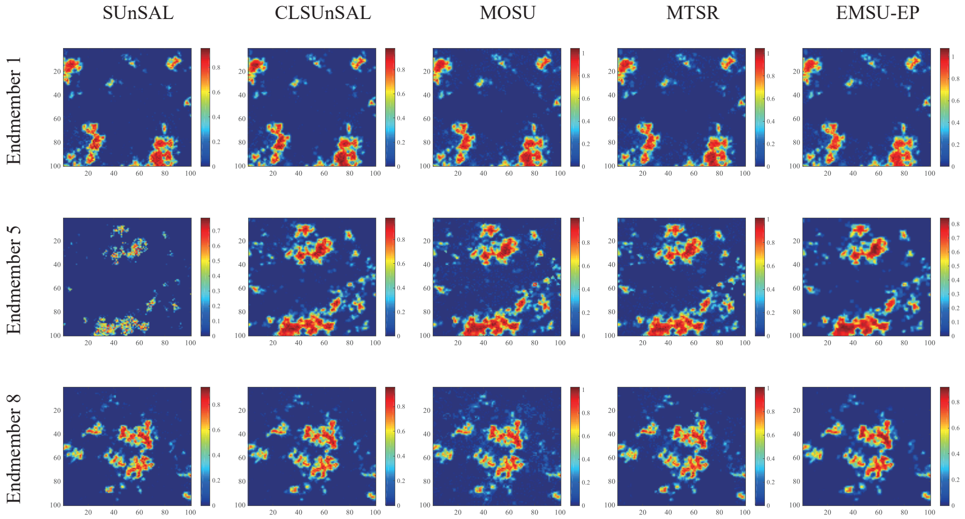


**Figure 9.** The estimated abundance maps of endmember 1, endmember 5 and endmember 8 for data 2 on 30 dB SNR obtained by different algorithms.

## 5. Conclusions

In this paper, we proposed an evolutionary multiobjective hyperspectral sparse unmixing algorithm with an endmember a priori strategy (EMSU-EP) to solve the large-scale hyperspectral sparse unmixing problem. EMSU-EP reconstructs the hyperspectral image by a single endmember to generate every endmember score first. Then the obtained endmember scores are used as prior knowledge to guide the generation of initial populations and new individuals. Experiments have demonstrated that the proposed EMSU-EP algorithm is effective in solving large-scale sparse unmixing problems, and EMSU-EP can maintain the superiority compared with the state-of-the-art sparse unmixing algorithms.

In the future, we will focus on reducing the noise on hyperspectral sparse unmixing problems and exploring the further improvement of EMSU-EP performance.

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

**Funding:** This work was supported by the National Natural Science Foundation of Shaanxi Province (grant no. 2021JQ-210), the Fundamental Research Funds for the Central Universities (Grant no. XJS200216), Key R & D programs of Shaanxi Province (Grant no. 2021ZDLGY02-06) and the National Natural Science Foundation of China (Grant no. 61973249).

**Data Availability Statement:** The data 1 and data 2 can be downloaded from http://levir.buaa.edu.cn/Code.htm (accessed on 22 August 2021).

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

## References

1. Shi, C.; Wang, L. Linear spatial spectral mixture model. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3599–3611. [CrossRef]
2. Nascimento, J.M.P.; Dias, J.M.B. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 898–910. [CrossRef]
3. Miao, L.; Qi, H. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 765–777. [CrossRef]
4. Zhou, G.; Xie, S.; Yang, Z.; Yang, J.; He, Z. Minimum-volume-constrained nonnegative matrix factorization: Enhanced ability of learning parts. *IEEE Trans. Neural Netw.* **2011**, *22*, 1626–1637. [CrossRef]
5. Li, J.; Bioucas-Dias, J.M.; Plaza, A.; Liu, L. Robust collaborative nonnegative matrix factorization for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6076–6090. [CrossRef]
6. Berman, M.; Kiiveri, H.; Lagerstrom, R.; Ernst, A.; Dunne, R.; Huntington, J.F. Ice: A statistical approach to identifying endmembers in hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 2085–2095. [CrossRef]
7. Candes, E.J.; Tao, T. Decoding by linear programming. *IEEE Trans. Inf. Theory* **2005**, *51*, 4203–4215. [CrossRef]
8. Bioucas-Dias, J.M.; Figueiredo, M.A. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In Proceedings of the 2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, Reykjavik, Iceland, 14–16 June 2010; pp. 1–4.
9. Iordache, M.; Bioucas-Dias, J.M.; Plaza, A. Collaborative sparse regression for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 341–354. [CrossRef]
10. Iordache, M.; Bioucas-Dias, J.M.; Plaza, A. Total variation spatial regularization for sparse hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 4484–4502. [CrossRef]
11. Zhong, Y.; Feng, R.; Zhang, L. Non-local sparse unmixing for hyperspectral remote sensing imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 1889–1909. [CrossRef]
12. Połap, D.; Woźniak, M. Red fox optimization algorithm. *Expert Syst. Appl.* **2021**, *166*, 114107. [CrossRef]
13. Połap, D.; Woźniak, M. Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism. *Symmetry* **2017**, *9*, 203. [CrossRef]
14. Khishe, M.; Mosavi, M. Chimp optimization algorithm. *Expert Syst. Appl.* **2020**, *149*, 113338. [CrossRef]
15. Gong, M.; Li, H.; Luo, E.; Liu, J.; Liu, J. A multiobjective cooperative coevolutionary algorithm for hyperspectral sparse unmixing. *IEEE Trans. Evol. Comput.* **2017**, *21*, 234–248. [CrossRef]
16. Jiang, X.; Gong, M.; Li, H.; Zhang, M.; Li, J. A two-phase multiobjective sparse unmixing approach for hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 508–523. [CrossRef]
17. Xu, X.; Shi, Z.; Pan, B. $l_0$-based sparse hyperspectral unmixing using spectral information and a multi-objectives formulation. *ISPRS J. Photogramm. Remote Sens.* **2018**, *141*, 46–58. [CrossRef]

18. Xu, X.; Shi, Z.; Pan, B.; Li, X. A classification-based model for multi-objective hyperspectral sparse unmixing. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9612–9625. [CrossRef]
19. Chang, C.I.; Heinz, D.C. Constrained subpixel target detection for remotely sensed imagery. *IEEE Trans. Geosci. Remote Sens.* **2000**, *38*, 1144–1159. [CrossRef]
20. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
21. Li, H.; Ong, Y.; Gong, M.; Wang, Z. Evolutionary multitasking sparse reconstruction: Framework and case study. *IEEE Trans. Evol. Comput.* **2019**, *23*, 733–747. [CrossRef]
22. Tian, Y.; Zhang, X.; Wang, C.; Jin, Y. An evolutionary algorithm for large-scale sparse multiobjective optimization problems. *IEEE Trans. Evol. Comput.* **2020**, *24*, 380–393. [CrossRef]
23. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
24. Branke, J.; Deb, K.; Dierolf, H.; Osswald, M. Finding knees in multi-objective optimization. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Birmingham, UK, 18–22 September 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 722–731.
25. Tian, Y.; Cheng, R.; Zhang, X.; Jin, Y. PlatEMO: A matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput. Intell. Mag.* **2017**, *12*, 73–87. [CrossRef]
26. Tang, W.; Shi, Z.; Wu, Y.; Zhang, C. Sparse unmixing of hyperspectral data using spectral a priori information. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 770–783. [CrossRef]
27. Li, H.; Zhang, Q.; Deng, J.; Xu, Z.-B. A preference-based multiobjective evolutionary approach for sparse optimization. *IEEE Trans. Neural Networks Learn. Syst.* **2018**, *29*, 1716–1731. [CrossRef]
28. Beume, N.; Naujoks, B.; Emmerich, M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **2007**, *181*, 1653–1669. [CrossRef]

electronics

MDPI

*Article*

# Evolutionary Convolutional Neural Network Optimization with Cross-Tasks Transfer Strategy

**Zhao Wang [1,2], Di Lu [2], Huabing Wang [1], Tongfei Liu [2] and Peng Li [2,*]**

[1] State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System (CEMEE), Luoyang 471003, China; wangzhao@xidian.edu.cn (Z.W.); xueshan@ustc.edu (H.W.)

[2] Key Laboratory of Electronic Information Countermeasure and Simulation Technology of Ministry of Education, Xidian University, No. 2 South TaiBai Road, Xi'an 710071, China; Di_Lu@stu.xidian.edu.cn (D.L.); liutongfei_home@hotmail.com (T.L.)

* Correspondence: li_peng001@163.com; Tel.: +86-137-0919-3246

**Abstract:** Convolutional neural networks (CNNs) have shown great success in a variety of real-world applications and the outstanding performance of the state-of-the-art CNNs is primarily driven by the elaborate architecture. Evolutionary convolutional neural network (ECNN) is a promising approach to design the optimal CNN architecture automatically. Nevertheless, most of the existing ECNN methods only focus on improving the performance of the discovered CNN architectures without considering the relevance between different classification tasks. Transfer learning is a human-like learning approach and has been introduced to solve complex problems in the domain of evolutionary algorithms (EAs). In this paper, an effective ECNN optimization method with cross-tasks transfer strategy (CTS) is proposed to facilitate the evolution process. The proposed method is then evaluated on benchmark image classification datasets as a case study. The experimental results show that the proposed method can not only speed up the evolutionary process significantly but also achieve competitive classification accuracy. To be specific, our proposed method can reach the same accuracy at least 40 iterations early and an improvement of accuracy for 0.88% and 3.12% on MNIST-FASHION and CIFAR10 datasets compared with ECNN, respectively.

**Keywords:** evolutionary algorithm; convolutional neural network; transfer learning; image classification

**Citation:** Wang, Z.; Lu, D.; Wang, H.; Liu, T.; Li, P. Evolutionary Convolutional Neural Network Optimization with Cross-Tasks Transfer Strategy. *Electronics* **2021**, *10*, 1857. https://doi.org/10.3390/electronics10151857

Academic Editor: Amir Mosavi

Received: 24 June 2021
Accepted: 29 July 2021
Published: 2 August 2021

## 1. Introduction

Machine learning has shown great success in various real-world applications. Among machine learning approaches, convolutional neural networks (CNNs), which show overwhelmingly superiority among machine learning approaches, have been widely used in various real-world applications, such as image processing [1], engineering [2], health care [3,4], and cognitive science [5], etc. Convolutional neural network commonly consists of convolution, pooling, and fully-connected layers and are trained on the source dataset and then applied to the target dataset. As is well-known, the success of CNNs mainly benefit from the improvement on fundamental CNN architectures, such as increasing the depth of neural networks, the employment of skip layers, and adding inner network structures, etc. However, the state-of-the-art CNN architectures with high performance are manually devised by experienced experts with trial-and-error. As designing efficient CNN architectures is a challenging process, researchers have developed algorithms to design the CNN architectures automatically, which aims to enhance the applicability and universality of CNNs.

Designing the best CNN architecture can be viewed as a neural architecture search (NAS) process on the given dataset, and the searching parameters include the number of convolutional layers, the configuration of different layers and the placement of skip layers, etc. In the domain of machine learning, reinforcement learning (RL) is an early method for NAS and is always collocated with recurrent neural network (RNN) to control the process of hyperparameter tuning [6]. In addition to reinforcement learning, Bayesian optimization [7] has also been widely used, which achieves global optimization by maintaining black-box functions that do not assume any specific forms. Some other machine learning approaches including grid search, random search and gradient search are explored in [8–10]. Nonetheless, with these methods remains the problem of taking up too much searching space, thus requiring excessive GPU memory.

Evolutionary algorithms (EAs) are optimization approaches which seek the optimal solution by simulating the natural evolution process inspired by Darwin's theory of evolution [11]. Evolutionary algorithm begins with generating individuals that undergo crossover, mutation, and selection to retain strong individuals and eliminate weak ones. After several generations, it can solve complex optimization problems and generate high-quality optimization schemes efficiently. Since evolutionary algorithm has flexible representations and strong searching capacity, it has been widely used to solve a variety of tasks, especially under the environment with non-convex or non-derivative functions [12–16]. In recent years, EAs have attracted great interest as they offer alternative methods to solve NAS problems. With flexible encoding strategy and strong searching capacity, EAs are becoming a promising approach to optimizing CNN architectures. The general framework of evolutionary convolutional neural network (ECNN) follows the evolution procedure, the performance of which mainly depends on the effective design of network architecture evolution strategy.

However, most of the ECNN methods focus on improving the performance of discovered CNN architectures without considering the relevance between different classification tasks. The evolution process has to start from scratch if the learning environment changes. In such a situation, evolutionary transfer learning which utilizes the knowledge from the previously solved tasks to facilitate solving target task, is promising for NAS problems. Evolutionary transfer learning is a human-like learning process, it has been frequently introduced to solve different but related problems for more effective evolutionary algorithms. In [17–22], EAs with transfer capacity are designed to solve extensive optimization problems like dynamic vehicle routing, heterogeneous and image classification problems, etc. With the assistance of knowledge from optimized solutions, the performance of evolutionary algorithms can be enhanced.

In this paper, an effective evolutionary convolutional neural network optimization method with cross-tasks transfer strategy (CTS-ECNN) is proposed to take full advantage of the knowledge extracted from the previously solved tasks when a new classification task is encountered. The main contributions of the proposed CTS-ECNN method are summarized as follows:

(1) We propose a simple and effective cross-tasks transfer strategy, which can select the valuable knowledge from the original task to transfer for improving the performance of the target task. Especially at the early generations, our method can increase the optimization speed significantly, which is important when the learning time or computing resource is limited;
(2) Within the case study of image classification tasks, it is demonstrated that the proposed CTS-ECNN can obtain better results than the ECNN that starts from scratch and some manually-designed state-of-the-art methods do;
(3) In the framework of the proposed CTS-ECNN, when a new task is encountered, we can extract knowledge from the optimized tasks. With more knowledge achieved from related tasks, the proposed method can be applied to more tasks rapidly without considering the sequence of tasks.

The rest of this paper is organized as follows: In Section 2, the related works are summarized. In Section 3, our method is introduced in detail. Section 4 gives the experimental settings and the analysis of the experimental results. Finally, the conclusions and future works are described in Section 5.

## 2. Related Works

### 2.1. Evolutionary CNN Optimization

Since the evolutionary algorithm has the advantage of gradient-free and being insensitive to local optimum, evolutionary deep learning has been an interesting domain recently. Among all of them, the neural network, especially CNN architecture optimization with the evolutionary algorithm, attracts much attention. The general framework of evolutionary convolutional neural network is concluded in Algorithm 1. As is shown in Algorithm 1, the whole process of evolutionary CNN follows the evolution procedure: initialization (step 1), evaluation (step 2), selection (step 4), mutation and crossover (step 5). Specially, the fitness evaluation in evolutionary CNN is executed by training the corresponding CNN architecture. The evaluation process is completed by optimizing the weights in CNN architecture to reach the maximal classification accuracy. To evaluate each individual's fitness accurately, a CNN is trained for several epochs by using the same initialization method, loss function and optimizer. The performance evaluation of CNN is provided in Algorithm 2.

**Algorithm 1** Evolutionary convolutional neural network.

```
Input: $N$: the max number of generations, $k$: the size of each generation, $p_m$: the mutation probability, $p_c$: the crossover probability.
Output: Individuals of the last generation with their fitness values.
1: Randomly initialize $k$ individuals and map them to the corresponding CNNs;
2: Compute the classification accuracy of each CNN to obtain the fitness value with Algorithm 2;
3: for $n = 1 : N$ do
4:     Generate a new generation with selecting method on parent individuals based on the fitness value;
5:     Produce offspring through the operator of mutation and crossover with probability $p_m$ and probability $p_c$;
6:     Compute the classification accuracy of each individual on offspring;
7: end for
```

**Algorithm 2** Performance evaluation of CNN.

```
Input: $p$: the individual, $D_{train}$: the training dataset, $D_{valid}$: the validation dataset, $T$: the epoch number, $B$: the training batch size, $L$: the loss function, $\eta$: the learning rate.
Output: The classification accuracy.
Map $p$ into the corresponding CNN architecture;
$\omega \leftarrow$ initialize the weights of CNN with predefined method;
for $t = 1 : T$ do
    $\eta \leftarrow$ update $\eta$ according to $t$;
    for each $B$ in $D_{train}$ do
        $\nabla\omega \leftarrow$ compute the gradient by $\partial L/\partial\omega$;
        $\omega \leftarrow \omega - \eta\nabla\omega$;
    end for
    Compute the classification accuracy on $D_{valid}$;
end for
```

In [23], Xie et al. introduced a binary encoding method to represent CNN and use GA to learn network architectures automatically. Concurrently, Cartesian genetic programming was used to design deep convolutional neural network architectures with a variable-length genotype-to-phenotype method in [24]. Real et al. [25] designed an image classifier named AmoebaNet which applies evolutionary algorithm to neural network topologies. Sun's work [26] gave a comprehensive comparison of manually designed and automatically designed neural networks, and then proposes CNN-GA which is competent for discovering deep neural networks. Lu and Whalen et al. [27] addressed multi-objective framework to neural architecture search, and the NSGANetV1 algorithm is demonstrated on new classification tasks, i.e., corrupted CIFAR-10, ImageNet-V2 and medical X-ray images.

### *2.2. Transfer Learning*

In the domain of machine learning, transfer learning has received significant interest on solving different but related problems for better effective deep learning performance. Knowledge transfer of internal representations is an example of transfer learning in [28]. A trained deep convolutional neural network is demonstrated that its components can be transferred to another network to learn new information with smaller training sets. Terekhov et al. [29] applied knowledge transfer to deep neural networks by re-using block-modular architecture to solve new tasks. This architecture can outperform networks trained from scratch and has fewer weights to learn. Based on the learning features of each layer in neural networks, Yosinski et al. [30] proposed a method which can quantify the transferability of layer features. The results show that transferability is affected by difficulties of splitting network layers and the specificity of higher network layers.

There is also a growing interest in evolutionary transfer algorithms in recent years. Evolutionary transfer algorithms have been applied to solving different types of problems, such as multi-task optimization, multi-objective optimization, and complex optimization, etc. In multi-task environment, [31] transferred knowledge through crossover based on the theory that the solving of one problem may facilitate the solving of other related problems. In [32], Y. Ong et al. extended knowledge transfer by designing a explicit auto-encoder to transfer optimized solutions instead of genetic crossover. In the domain of multi-objective optimization, Liang et al. [18] devised a one-layer auto-encoder to enhance the performance of evolutionary algorithm by transferring knowledge across heterogeneous problems. Iqbal et al. [19] developed the GP-criptor to transfer learning GP-criptor which can reuse knowledge from past solved classification problems to improve image classification accuracy.

In this work, we focus on utilizing transfer learning to improve the performance and efficiency of neural architecture evolution. The proposed method can take full advantage of knowledge transferred from the previous solved tasks when a new classification task is encountered.

## 3. Materials and Method

As mentioned before, the key problem of transfer learning in EAs is how to utilize the knowledge from the source tasks efficiently. Which information to be transferred and how to transfer the knowledge determine whether the transfer process can facilitate the evolution of target task better than random initialization. In this section, we will introduce the CTS-ECNN method which constructs the suitable individuals for transference based on the multi-population framework [33]. Meanwhile, considering the high computation cost of evaluating the performance of a CNN architecture, we also use a clustering method to accelerate the transfer process. Figure 1 shows the workflow of the CTS, as well as the associated ECNN.

As shown in Figure 1, the proposed CTS-ECNN method is composed of three modules, i.e., classification tasks, evolutionary algorithm of CNN, and cross-tasks transfer. First, a set of individuals representing different CNN architectures are evolved on the first classification task with their performances (shown in Section 3.1). After evolution, the CNN architectures with top fitness values are encoded for clustering similar individuals with affinity propagation (AP) method (shown in Section 3.2). With clustered individuals, exemplars of each cluster are evaluated on the target classification task, and then their performances are used as the ranking index prepared for constructing the subpopulation (shown in Section 3.3). When the mixed subpopulation is constructed, we can transfer the useful knowledge to the target classification task through applying the selected individuals as the initial generation. To clarify the proposed method, we give a general framework of CTS-ECNN in Algorithm 3. Noting that our proposed method provides a sequential transfer framework applicable to different classification tasks. If we have discovered CNN architectures on the preceding ($M$-1) tasks, the $M$-*th* task can be facilitated by utilizing the knowledge obtained from the previously solved ($M$-1) tasks.

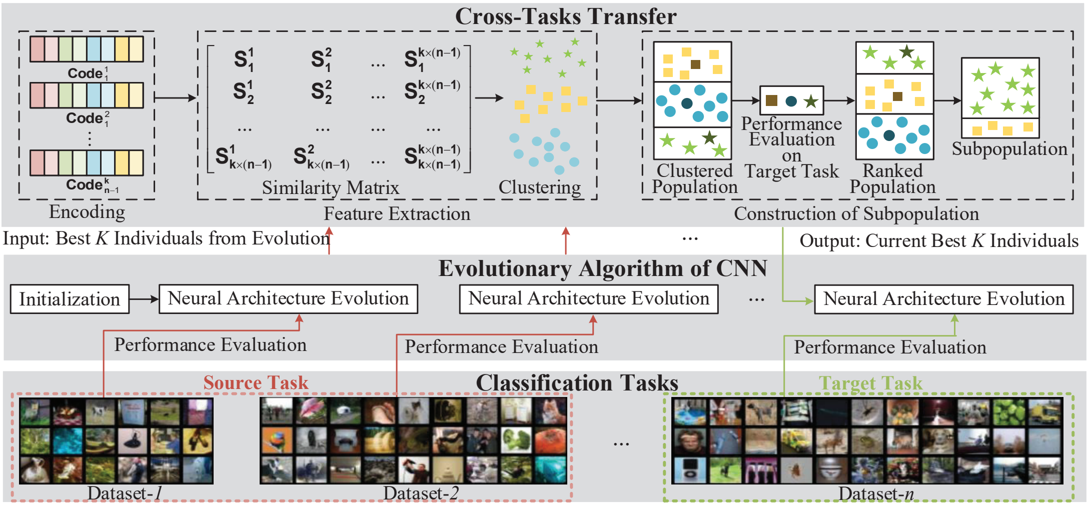

Cross-Tasks Transfer
Code$_1^1$
Code$_1^2$
Code$_{n-1}^k$
Encoding
$S_1^1$ $S_1^2$ ... $S_1^{k\times(n-1)}$
$S_2^1$ $S_2^2$ ... $S_2^{k\times(n-1)}$
$S_{k\times(n-1)}^1$ $S_{k\times(n-1)}^2$ ... $S_{k\times(n-1)}^{k\times(n-1)}$
Similarity Matrix
Clustering
Feature Extraction
Clustered Population
Performance Evaluation on Target Task
Ranked Population
Subpopulation
Construction of Subpopulation
Input: Best K Individuals from Evolution
Output: Current Best K Individuals
Evolutionary Algorithm of CNN
Initialization
Neural Architecture Evolution
Neural Architecture Evolution
Neural Architecture Evolution
Performance Evaluation
Performance Evaluation
Performance Evaluation
Classification Tasks
Source Task
Target Task
Dataset-1
Dataset-2
Dataset-n

**Figure 1.** The workflow of the CTS-ECNN.

**Algorithm 3** The pseudocode of CTS-ECNN.

**Input:** $N$: the max number of generation, $k$: the size of each generations, $D_M$: the target dataset.

**Output:** The best CNN architecture for $D_M$.

```
1:  for $m = 1 : M$ do
2:    if $m = 1$ then
3:      $\{O_P\}_k \leftarrow$ Randomly initialize $k$ individuals;
4:    else
5:      $\{O_P\}_k \leftarrow$ Transfer $k$ individuals based on Algorithm 4;
6:    end if
7:    Set $n \leftarrow 0$;
8:    while $n \leq N$ do
9:      Obtain offspring $\{O_C\}_k$ with crossover and mutation operators;
10:     $\{O_R\}_{2k} \leftarrow \{O_P\}_k \cup \{O_C\}_k$;
11:     Evaluate each individual in $\{O_R\}_{2k}$;
12:     $\{O_P\}_k \leftarrow$ Select the next generation;
13:     $n \leftarrow n + 1$;
14:   end while
15:   while $m < M$ do
16:     $\{O_r^c\} \leftarrow$ collect individuals with top fitness values for CTS in Algorithm 4;
17:   end while
18: end for
```

### 3.1. Neural Architecture Evolution

Generally, the whole process of ECNN follows the procedure in Algorithm 3 (steps 1–14). The first step is to design a proper genotype-to-phenotype mapping strategy. We provide a variable-length string representation to describe the CNN architecture. In our method, we prepare two types of convolutional and pooling operators for building a CNN architecture, i.e., the standard and residual convolutional operators, the max and average pooling operators. The max number of convolutional operators $N_c$ and pooling operators

$N_p$ are predefined. It is noted that the proposed method focuses on the optimization of neural network structure, so we select these building blocks as functional nodes to realize flexible genotype-to-phenotype mapping. For the first classification task, a set of individuals with predefined population size is randomly initialized based on the string representation. Accordingly, subsequent classification tasks use the assigned subpopulation from cross-tasks transfer as the initial generation (steps 2–6).

During evolution, mutation and crossover operator are implemented on each generation. When $\{O_R\}_{2k}$ is obtained, each produced individual is evaluated on the corresponding dataset to compute its classification accuracy by training the CNN architecture it represents through several epochs, and then these accuracies serve as fitness values to produce a new generation. When the max number of generations is reached, we not only obtain the optimized CNN architecture for the current classification task but also utilize individuals with top fitness values as the learning resource to facilitate posterior classification tasks.

### *3.2. Encoding and Extraction of Feature*

The encoding operator acts on the optimized CNN architectures which have been evolved on previous classification tasks. To extract the features of each CNN architecture, these architectures have to be encoded into fixed-length strings which are suitable for similarity computation. Each convolutional operator is encoded into a quaternion as (*type*, *channel*, *filter*, *in*), where *type* for the standard and residual convolutional operator is set to 0 and 1, respectively. As for pooling operators, each of them is encoded into a pair as (*type*, *in*), *type* of the max and average pooling operators is denoted by 0 and 1. There is no need to encode the parameter of *out* because it can be deduced by the *in* of subsequent operator. When the number of convolutional operator is smaller than $N_c$ or the number of pooling operators is smaller than $N_p$, the blank position will be set to zeros to keep the coding length invariable.

As the encoding of CNN architectures has been finished, the codes containing structure information can be used directly by similarity computation. Algorithm 4 shows the framework of our proposed CTS, and steps 2–7 give a general process of constructing a similarity matrix. We use Euclidean distance as the similarity of CNN architectures:

$$s(i,j) = -\|x_i - x_j\|^2 \tag{1}$$

where $x_i$ and $x_j$ are two different codes, and the setting of a negative squared error is for convenient calculation. Each code of its corresponding CNN architecture is compared with all the other codes string-to-string to work out the similarity *s(i, j)*, and then the similarity *s(i, i)* for each code is set to a shared value, for which we choose the median of the input similarity.

Considering the demand of reducing transfer computation and utilizing knowledge from previous tasks efficiently, clustering is a necessary preprocessing for the construction of subpopulation. Of all the clustering methods, AP has the advantages of good robustness and accuracy over other clustering methods [34]. It does not need to determine the number of clusters before running the algorithm and is based on the concept of "message passing" between data points, which updates two matrices:

$$r(i,j) \leftarrow s(i,j) - \max_{j' \neq j}\{a(i,j') + s(i,j')\} \tag{2}$$

$$a(i,j) \leftarrow \min\left\{0, r(j,j) + \sum_{i' \notin \{i,j\}} \max\{0, r(i',j)\}\right\} \tag{3}$$

where *r(i,j)* is the value of responsibility matrix and reflects the fitness that *j* serves as the exemplar for *i* on account of the other potential exemplars. *a(i,j)* is the value of availability matrix and reflects the appropriateness that *i* chooses *j* as its exemplar on account of the other potential exemplars. To be specific, *a(i,j′)* represents the belongingness of other points to *i* except *j*. *s(i,j′)* represents the attraction of other points to *i* except *j*. If the value of *r(i,j)* is greater than 0, it means that *j* has better chance to become the exemplar; *r(i,j')* represents the similarity that *j* becomes the exemplar of other points except *i*. Taking all the attraction values greater than or equal to 0 and the possibility that *j* is the exemplar into consideration, *a(i,j)* represents the cumulative proof that *i* chooses *j* as the exemplar.

Both matrices are initialized to all zeroes. Iterations proceed until cluster boundaries remain unchanged over several iterations. The *j* with the maximum value of *a(i,j) + r(i,j)* will be chosen as the exemplar of its corresponding cluster. When AP terminates, the number of clusters and the exemplar of each cluster are obtained.

**Algorithm 4** Cross-task transfer strategy (CTS).

```
Input: $\{O_r^c | r = 1, 2, ..., k, c = 1, 2, ..., M-1\}$: individuals with top fitness values from the preceding $(M-1)$ datasets, $q_m$: the mutation probability, $TP$: the transfer parameter.
Output: The parent $P_{Trans}$ for $D_M$.
$E_t \leftarrow$ encode $O_r^c$ independently;
if $i \neq j$ then
    $s(i, j) \leftarrow$ compute the similarity between $E_i$ and $E_j$;
end if
if $i = j$ then
    $s(i, i) \leftarrow$ input the median of the acquired similarities;
end if
Cluster CNN architectures with $AP$ algorithm;
$p \leftarrow$ the number of clusters;
$\{M\}_p \leftarrow$ choose the exemplar of each cluster;
Approximate the classification accuracies of $\{M\}_p$;
$P_{opt} \leftarrow$ choose the optimal cluster;
$N_{sub} \leftarrow$ determine the size of suboptimal subpopulation with $TP$;
$P_{sub} \leftarrow$ randomly choose $N_{sub}$ individuals from the suboptimal cluster;
if $N_{opt} + N_{sub} < k$ then
    $P_{ext} \leftarrow$ perform mutation operator with probability $q_m$ on $P_{opt}$;
    $P_{Trans} \leftarrow P_{opt} \cup P_{sub} \cup P_{ext}$;
else if $N_{opt} + N_{sub} > k$ then
    $P_{opt} \leftarrow$ choose $(k - N_{sub})$ individuals from $P_{opt}$ randomly;
    $P_{Trans} \leftarrow P_{opt} \cup P_{sub}$;
else
    $P_{Trans} \leftarrow P_{opt} \cup P_{sub}$;
end if
```

### *3.3. Construction of Subpopulation*

In original transfer learning strategies, the entire knowledge extracted from previous tasks is evaluated on the target task to sort out the best solution. However, considering the high computational cost of evaluating the performance of a CNN architecture, it is computationally cumbersome to evaluate each individual especially when we have numerous source tasks for transfer learning. In our method, we only evaluate the exemplar of each cluster on the target task to represent its corresponding cluster. The proposed CTS is a straightforward way to utilize the ranked clusters of alternative CNN architectures based on exemplars' performance. To enable efficient transfer, we construct the subpopulation by exploiting high-quality individuals and exploring new individuals.

The efficacy of the proposed CTS-ECNN depends on how the subpopulation of each target task is constructed. On account of the complementarity of individuals, we adopt the multi-population framework to construct subpopulation which includes the optimal cluster $P_{opt}$ and the suboptimal cluster $P_{sub}$. Specifically, $P_{opt}$ can be viewed as the individuals which carry most problem-solving knowledge and thus can be viewed as the individuals belonging to target task, $P_{sub}$ can be viewed as the extra individuals selected from related tasks accordingly. When the mixed subpopulation is constructed, it can collect inter-task knowledge and inner-task knowledge to generate offspring for the target task. $N_{sub}$

is determined by the cross-task transfer parameter $TP$, thus the number of suboptimal individuals $N_{sub}$ is denoted by

$$N_{sub} = \min\{[TP \times k], N_{sub,max}\} \tag{4}$$

where $k$ represents the size of each generation and $N_{sub,max}$ is the size of suboptimal cluster which guarantees that cross-task knowledge will not overflow. $TP$ is the transfer parameter which controls the degree of inter-task knowledge transfer thus maintaining the balance of subpopulation.

It is an important issue to set the transfer parameter $TP$ thoughtfully as it controls the amount of inner-task and inter-task knowledge transferred into the target task. To be specific, if $TP$ is large, the more extra individuals from related tasks are collected, therefore the more inter-task knowledge can be transferred to target task. Correspondingly, if $TP$ is small, the more inner-task knowledge will be extracted. To conclude, a large $TP$ is suitable for a compact searching space where individuals have small divergence, while a small $TP$ is suitable for individuals with significantly different performances.

With $N_{sub}$ determined by $TP$, we randomly choose $N_{sub}$ individuals in the suboptimal cluster as a part of subpopulation. In Algorithm 4, the construction of subpopulation is executed in step 15–23. As shown in Algorithm 4, the rest part of the subpopulation are produced by the optimal cluster. When $N_{opt}$ is insufficient, some extra individuals are produced by operating mutation on the individuals of optimal cluster with probability $q_m$ to keep the size of generation invariable. On the contrary, if ($N_{opt}$ + $N_{sub}$) exceeds the size of generation, we randomly choose ($k - N_{sub}$) individuals from the optimal cluster for the same purpose.

The constructed subpopulation is used as the current best generation for the target classification task to accelerate the evolution of CNN architectures. Particularly, the more source tasks we have trained, the more transfer knowledge we can utilize to facilitate the subsequent tasks. More importantly, as the knowledge from all the previous tasks participates in our cross-task transfer process, the sequential transfer process can keep going without deploying specified order of tasks.

### *3.4. Training and Prediction*

Based on conventional settings in machine learning community, stochastic gradient descent (SGD) with a batch size of 128 is used to train the CNN architectures, whose weights are initialized by He's method [35]. For fear of over-fitting, the weight decay is set to $5 \times 10^{-4}$. The learning rate is initialized to $10^{-2}$ for the first 30 epochs, followed by $10^{-3}$ for 120 epochs, $10^{-4}$ for 90 epochs, and $10^{-5}$ for 30 epochs. Each CNN architecture is trained for 50 epochs in the phase of fitness evaluation to reduce computation time. All of the CNN architectures are trained on two NVIDIA 1080TI GPUs. Noting that these parameter settings are applied in both experimental scenarios.

For the parameter settings of neural architecture evolution, the max number of convolutional operators $N_c$ and pooling operators $N_p$ are set to 10 and 5, respectively. We set the mutation probability $p_m$ and the crossover probability $p_c$ to 0.8 and 0.2, respectively, to accelerate the evolution of new CNN architectures. The number of generations for the target classification dataset is set to 50 and each generation contains 20 individuals. Note that the max generation for the source classification dataset which is prepared for the CTS is set as 20.

## **4. Experimental Results and Discussion**

### *4.1. Datasets*

In the case study, the proposed CTS-ECNN method, which is based on transfer learning, is evaluated using two benchmark image classification datasets. Results are compared with the ECNN that starts from scratch and some state-of-the-art methods.

As mentioned in Section 3, the proposed method is a sequential transfer framework, so we design two experimental scenarios to test its applicability for different classification tasks. For the first scenario, our method is tested on the MNIST-FASHION dataset and the transfer process is based on the MNIST classification task which has been optimized before. For the second scenario, our method is tested on the CIFAR10 dataset and the transfer process is based on the MNIST and MNIST-FASHION classification task which have been optimized before.

The MNIST dataset is a large database of handwritten digits, which consists of 60,000 grayscale images for training and 10,000 grayscale images for testing, and each of them has the dimension of 28 × 28. There are 10 categories, i.e., digits from 0 to 9, which have the equal number of samples for both the training and testing set. The MNIST-FASHION dataset shares the same image size and structure of training and testing splits with the MNIST dataset. The only difference between them is that there are 10 categories of commodity in MNIST-FASHION. The CIFAR10 dataset is a subset of 80 million tiny images, which consists of 50,000 color images for training and 10,000 color images for testing, and each of them has the dimension of 32 × 32. There are also 10 categories, which have the equal number of samples for both the training and testing set. As there are no validation sets in these benchmark datasets, 10% images of the training sets will be randomly selected as the validation sets to attain the fitness value.

### *4.2. Results of the First Experimental Scenario*

We first compare the classification accuracy of each evolution step for the proposed CTS-ECNN method and the ECNN that starts from scratch on the MNIST-FASHION testing dataset. The maximum classification accuracy of each evolution process is visualized in Figure 2. It is obvious that the max classification accuracy is improved significantly in the first few generations with the transferred knowledge from the MNIST classification task. As shown in Figure 2, the proposed CTS-ECNN can achieve the same accuracy about 40 iterations early compared with ECNN that starts from scratch. When the evolution terminates, our method still obtains better classification accuracy than the original ECNN method does. The above result shows that the proposed CTS-ECNN can achieve the valuable knowledge from the MNIST classification task to help the neural network optimization of MNIST-FASHION classification task.

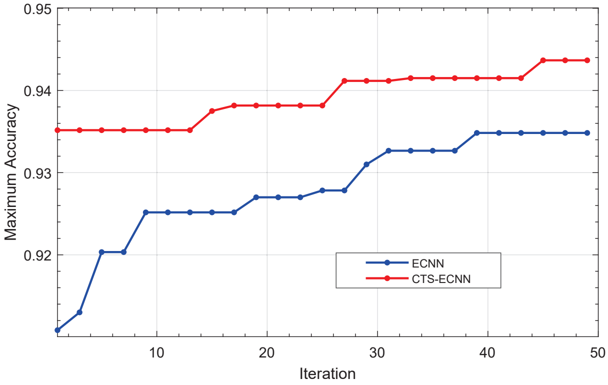


**Figure 2.** The maximum classification accuracy of each evolution step on the MNIST-FASHION testing dataset.

To make a comprehensive comparison of the two methods, classification performance of all the individuals in each evolution step are reported in Figure 3. As shown in Figure 3, the red and blue points represent the individuals in CTS-ECNN method and the original ECNN method that starts from scratch, respectively. As can be seen in each iteration, most of the individuals of our method can obtain better performance, which means with transferred knowledge the excellent parent generation is more likely to generate good offspring via evolution process. There are also a few red points lying below all the blue points in the figure, especially at the early iterations of the evolution process. As evolutionary algorithm is a heuristic search method that uses crossover and mutation operators to generate new populations, good individuals may also produce poor offspring. However, the historically best individuals are always maintained, so the evolutionary process can keep a steady improvement. With the evolution proceeding, individuals of our method perform better aggregation index than the original ECNN method does. We argue that the evolutionary algorithm tends to preserve the useful transferred knowledge, so that the excellent individuals are more easily to generate.

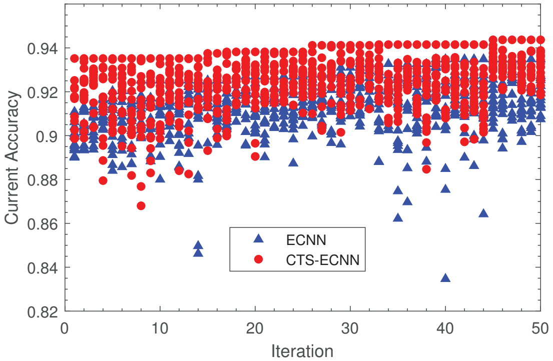


**Figure 3.** The current classification accuracy of each evolution step on the MNIST-FASHION testing dataset.

In addition to reporting the overall results of each iteration, we complete quantitative analysis on these two methods to obtain precise statistics. Results are summarized in Table 1. With the help of CTS, we can always find better network architectures which can achieve better classification accuracy. After the 10th generation, classification accuracies of the two methods keep a similar rate of growth and the accuracy improvement becomes smaller than the previous generations. To be specific, among the 10th and 50th generation, the proposed CTS-ECNN and original ECNN attain the improvement of 0.85% and 0.87% on maximum classification accuracy, respectively. Although during the period of evolution the average and medium classification accuracy are fluctuating a little, on the whole they gradually get higher. According to three kinds of difference value, i.e., the maximum, the average and the medium accuracy difference, the CTS-ECNN method shows significant advantage over the ECNN algorithm. This is important, because it means the CTS-ECNN algorithm can guarantee the overall improvement on original ECNN algorithms. After 50 generations, the maximum classification accuracy of the CTS-ECNN method reaches 94.37% and keeps a leading margin of 0.88% over the original ECNN method.

### *4.3. Results of the Second Experimental Scenario*

As mentioned before, we have two optimized classification tasks, i.e., the MNIST and MNIST-FASHION dataset, which means when a new task is encountered, we can extract knowledge from the above three tasks. We will first compare the CTS-ECNN method based on the two classification tasks with the original ECNN method. The maximum classification accuracy of the two methods on the CIFAR10 testing dataset is shown in Figure 4. As in the MNIST-FASHION experiment, the maximum classification accuracy grows from generation to generation. Compared with the first experiment on MNIS-FASHION dataset, with more knowledge extracted from the previously solved two classification tasks, the initial generation of CTS-ECNN can outperform even the last generation of the original ECNN. This means our method can not only skip the process of random initialization but also generate better individuals. In the end of the evolution process, our proposed method obtains the improvement of about 3% for the maximum classification accuracy, which can demonstrate the scalability of our proposed method. As we all know that the iteration of the neural network optimization is time-consuming, the experiment with 50 generations can demonstrate that our method has the ability to reduce the computational cost.

**Table 1.** Classification accuracy on the MNIST-FASHION testing dataset. Diff is the difference of classification accuracy with ECNN that starts from scratch. Gen represents different evolution steps.

| Gen | Max % | Diff % | Avg % | Diff % | Med % | Diff % |
|---|---|---|---|---|---|---|
| 01 | 93.52 | 2.43 | 91.49 | 1.72 | 91.73 | 2.13 |
| 05 | 93.52 | 1.48 | 91.76 | 1.26 | 91.67 | 0.73 |
| 10 | 93.52 | 1.00 | 91.72 | 1.17 | 91.95 | 1.03 |
| 15 | 93.75 | 1.23 | 91.93 | 0.37 | 91.93 | 0.38 |
| 20 | 93.81 | 1.12 | 92.12 | 1.06 | 92.45 | 1.52 |
| 25 | 93.81 | 1.03 | 92.70 | 1.10 | 92.89 | 1.33 |
| 30 | 94.12 | 1.01 | 92.58 | 0.60 | 92.42 | 0.30 |
| 35 | 94.15 | 0.88 | 92.20 | 1.61 | 92.13 | 0.72 |
| 40 | 94.15 | 0.67 | 92.35 | 1.79 | 92.50 | 1.33 |
| 45 | 94.37 | 0.88 | 93.38 | 1.78 | 93.28 | 1.73 |
| 50 | 94.37 | 0.88 | 93.10 | 1.22 | 93.13 | 1.38 |

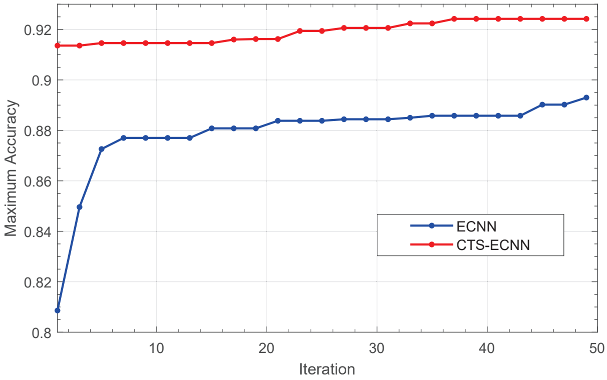


**Figure 4.** The maximum classification accuracy of each evolution step on the CIFAR10 validation dataset.

In order to better understand the details of the proposed method on the CIFAR10 classification task, we draw box plots in Figure 5. As is shown in Figure 5, both of the two methods show increase for the maximum classification accuracy and fluctuation for the median classification accuracy. However, our proposed method shows smaller fluctuation and better median classification accuracy. By investigating the height of each box, it can also be observed that the variation of the classification accuracy during each generation of our proposed method is much smaller than the original ECNN, which implies the evolution processes towards a more steady state on the CIFAR10 classification task with transferred knowledge.

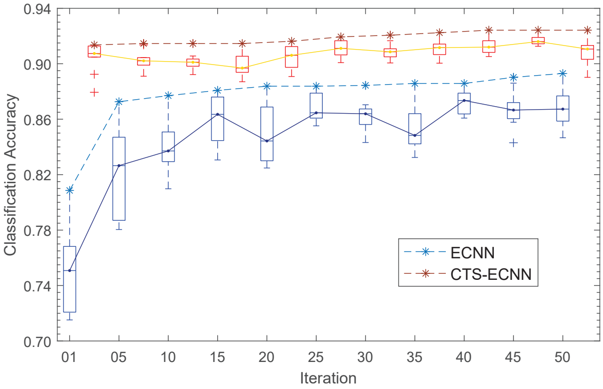


**Figure 5.** The classification accuracy of different evolution steps on the CIFAR10 testing dataset. The red box represents CTS-ECNN and blue box represents the original ECNN. The max and median classification accuracy of different evolution steps are connected by a dashed line and solid line.

In addition, our proposed method is compared with some state-of-the-art methods in Table 2. We group these methods into two different categories, namely manually-designed methods and automatically-designed methods. For the first category including ResNet (depth = 1.10), ResNet (depth = 1.202) [36], Maxout [37], Network in Network [38] and Highway Network [39], we mainly compare the classification accuracy. Among these manually-designed peer competitors, after the 50th generation, our proposed method can obtain higher classification accuracy than most of the state-of-the-art CNN architectures but lower than ResNet (depth = 101). We note that ResNet (depth = 101) is much deeper, i.e., ResNet (depth = 101) has 101 layers while the proposed CTS-ECNN has less than 15 layers (10 convolutional layers and 5 pooling layers). Even at the 10th generation, our method can outperform Maxout and Network in Network. It can be demonstrated that when compared with the state-of-the-art manually designed methods, the proposed method can design competitive CNN architecture automatically with limit computation cost. For the second category including hierarchical evolution [40], CGP-CNN [24], genetic CNN [23], and the proposed CTS-ECNN, we compare the classification accuracy and the number of iterations that each method costs, which represent the efficiency of each method. Among these automatically-designed peer competitors, hierarchical evolution and CGP-CNN obtain 3.96% and 1.6% improvements on the CIFAR10 testing dataset over our proposed method. However, both of the methods consume much more iterations to obtain the best classification accuracy and hierarchical evolution focus on convolutional cells, rather than the entire neural network architecture [40]. It is noted that the classification accuracy of genetic CNN is slightly higher than CTS-ECNN, but it still requires the manual tuning based on expertise. It is demonstrated that our method could find competitive CNN architecture with limited computational resources. It makes sense, as someone with little knowledge of neural network architecture can design a competent neural network to solve the certain task easily.

**Table 2.** The comparisons between the proposed method and the state-of-the-art methods in terms of the classification accuracy (%) on the CIFAR10 testing dataset. Gen represents the evolution steps each method takes.

| | **Method** | **Acc %** | **Gen** |
|---|---|---|---|
| Manually Designed | ResNet (depth = 101) | 93.57 | – |
| | ResNet (depth = 1202) | 92.07 | – |
| | Maxout | 90.70 | – |
| | Network in Network | 91.19 | – |
| | Highway Network | 92.40 | – |
| Automatically Designed | Hierarchical Evolution | 96.37 | 7000 |
| | CGP-CNN | 94.02 | 300 |
| | Genetic CNN | 92.90 | 50 |
| | **CTS-ECNN** (G-10) | 91.46 | 10 |
| | **CTS-ECNN** (G-30) | 92.06 | 30 |
| | **CTS-ECNN** (G-50) | 92.42 | 50 |

## 5. Conclusions

In this paper, we apply the transfer learning to facilitate the evolutionary CNN architecture optimization. We propose an effective ECNN method with cross-task transfer strategy named CTS-ECNN which constructs the suitable individuals to transfer without taking up too much computational resource. For the case study, our proposed method is compared with the original ECNN and some state-of-the-art methods on benchmark image classification datasets. The results show that our method can not only accelerate the evolution process significantly but also find competitive CNN architectures.

However, our method still suffers from several drawbacks. First, although we attempt to reduce the computational cost of transfer strategy, the process of neural architecture evolution on source tasks also requires computational resource. Second, in this work, transfer strategy is only applied to the initial generation. It would be interesting to transfer knowledge among each generation. The above directions are left for future work.

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

**Funding:** This work was supported by the National Natural Science Foundation of Shaanxi Province (grant no. 2021JQ-210), the Fundamental Research Funds for the Central Universities (Grant no. XJS200216, JB210202) and the National Natural Science Foundation of China (Grant no. 62036006).

**Data Availability Statement:** The MNIST dataset can be downloaded from http://yann.lecun.com/exdb/mnist/ (accessed on 3 July 2021). The MNIST-FASHION dataset can be downloaded from https://github.com/zalandoresearch/fashion-mnist (accessed on 3 July 2021). The CIFAR10 dataset can be downloaded from https://www.kaggle.com/c/cifar-10 (accessed on 3 July 2021).

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

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| CNNs | Convolutional neural networks |
| ECNN | Evolutionary convolutional neural network |
| EAs | Evolutionary algorithms |
| CTS | Cross-tasks transfer strategy |
| NAS | Neural architecture search |
| RL | Reinforcement learning |
| RNN | Recurrent neural network |
| AP | Affinity propagation |
| SGD | Stochastic gradient descent |

## References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
2. Jamshidi, M.; Lalbakhsh, A.; Lotfi, S.; Siahkamari, H.; Mohamadzade, B.; Jalilian, J. A neuro-based approach to designing a wilkinson power divider. *Int. J. Microw. Comput. Aided Eng.* **2020**, *30*, e22091. [CrossRef]
3. Roshani, S.; Jamshidi, M.B.; Mohebi, F.; Roshani, S. Design and modeling of a compact power divider with squared resonators using artificial intelligence. *Wirel. Pers. Commun.* **2021**, *117*, 2085–2096. [CrossRef]
4. Jamshidi, M.B.; Lalbakhsh, A.; Talla, J.; Peroutka, Z.; Roshani, S.; Matousek, V.; Roshani, S.; Mirmozafari, M.; Malek, Z.; Spada, L.L.; et al. Deep learning techniques and covid-19 drug discovery: Fundamentals, state-of-the-art and future directions. *Emerg. Technol. During Era COVID Pandemic* **2021**, *348*, 9.
5. Jamshidi, M.B.; Alibeigi, N.; Rabbani, N.; Oryani, B.; Lalbakhsh, A. Artificial neural networks: A powerful tool for cognitive science. In Proceedings of the 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 1–3 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 674–679.
6. Bello, I.; Zoph, B.; Vasudevan, V.; Le, Q.V. Neural optimizer search with reinforcement learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 459–468.
7. Snoek, J.; Larochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. In Proceedings of the Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012.
8. Pontes, F.; Amorim, G.; Balestrassi, P.; Paiva, A.; Ferreira, J. Design of experiments and focused grid search for neural network parameter optimization. *Neurocomputing* **2016**, *186*, 22–34. [CrossRef]
9. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
10. Bengio, Y. Gradient-based optimization of hyperparameters. *Neural Comput.* **2000**, *12*, 1889–1900. [CrossRef] [PubMed]
11. Bäck, T.; Schwefel, H.-P. An overview of evolutionary algorithms for parameter optimization. *Evol. Comput.* **1993**, *1*, 1–23. [CrossRef]
12. Ramakurthi, V.B.; Manupati, V.; Machado, J.; Varela, L. A hybrid multi-objective evolutionary algorithm-based semantic foundation for sustainable distributed manufacturing systems. *Appl. Sci.* **2021**, *11*, 6314. [CrossRef]
13. Abualigah, L.; Diabat, A.; Sumari, P.; Gandomi, A.H. A novel evolutionary arithmetic optimization algorithm for multilevel thresholding segmentation of covid-19 ct images. *Processes* **2021**, *9*, 1155. [CrossRef]
14. Guerraiche, K.; Dekhici, L.; Chatelet, E.; Zeblah, A. Multi-objective electrical power system design optimization using a modified bat algorithm. *Energies* **2021**, *14*, 3956. [CrossRef]
15. Yılmaz, E.M.; Güntert, P.; Etaner-Uyar, Ş. Evaluation of multi-objective optimization algorithms for nmr chemical shift assignment. *Molecules* **2021**, *26*, 3699. [CrossRef]
16. Ponti, A.; Candelieri, A.; Archetti, F. A new evolutionary approach to optimal sensor placement in water distribution networks. *Water* **2021**, *13*, 1625. [CrossRef]
17. Zhou, L.; Feng, L.; Gupta, A.; Ong, Y.; Liu, K.; Chen, C.; Sha, E.; Yang, B.; Yan, B.W. Solving dynamic vehicle routing problem via evolutionary search with learning capability. In Proceedings of the IEEE Congress on Evolutionary Computation, San Sebastián, Spain, 5–8 June 2017; pp. 890–896.
18. Feng, L.; Ong, Y.; Jiang, S.; Gupta, A. Autoencoding evolutionary search with learning across heterogeneous problems. *IEEE Trans. Evol. Comput.* **2017**, *21*, 760–772. [CrossRef]
19. Iqbal, M.; Xue, B.; Al-Sahaf, H.; Zhang, M. Cross-domain reuse of extracted knowledge in genetic programming for image classification. *IEEE Trans. Evol. Comput.* **2017**, *21*, 569–587. [CrossRef]
20. Xu, Q.; Wang, N.; Wang, L.; Li, W.; Sun, Q. Multi-task optimization and multi-task evolutionary computation in the past five years: A brief review. *Mathematics* **2021**, *9*, 864. [CrossRef]
21. Dumitru, D.; Dioșan, L.; Andreica, A.; Bálint, Z. A transfer learning approach on the optimization of edge detectors for medical images using particle swarm optimization. *Entropy* **2021**, *23*, 414. [CrossRef]
22. Chu, S.-C.; Zhuang, Z.; Li, J.; Pan, J.-S. A novel binary quasi-affine transformation evolutionary (quatre) algorithm. *Appl. Sci.* **2021**, *11*, 2251. [CrossRef]

23. Xie, L.; Yuille, A. Genetic CNN. In Proceedings of the IEEE International Conference on Computer Vision ICCV, Venice, Italy, 22–29 October 2017; pp. 1388–1397.
24. Suganuma, M.; Shirakawa, S.; Nagao, T. A genetic programming approach to designing convolutional neural network architectures. In Proceedings of the Genetic and Evolutionary Computation Conference, Berlin, Germany, 15–19 July 2017; pp. 497–504.
25. Real, E.; Aggarwal, A.; Huang, Y.; Le, Q. Regularized evolution for image classifier architecture search. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4780–4789.
26. Sun, Y.; Xue, B.; Zhang, M.; Yen, G.; Lv, J. Automatically designing cnn architectures using the genetic algorithm for image classification. *IEEE Trans. Cybern.* **2020**, *50*, 3840–3854. [CrossRef] [PubMed]
27. Lu, Z.; Whalen, I.; Boddeti, V.; Dhebar, Y.; Deb, K.; Goodman, E.; Banzhaf, W. Nsga-net: Neural architecture search using multi-objective genetic algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference, Prague, Czech Republic, 13–17 July 2019; pp. 419–427.
28. Gutstein, O.F.S.; Freudenthal, E. Knowledge transfer in deep convolutional neural nets. *Int. J. Artif. Intell. Tools* **2008**, *17*, 555–567. [CrossRef]
29. Terekhov, A.V.; Montone, G.; O'Regan, J. Knowledge transfer in deep block-modular neural networks. *arXiv* **2015**, arXiv:abs/1908.08017.
30. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How Transferable Are Features in Deep Neural Networks? In *Advances in Neural Information Processing Systems 27*; Curran Associates, Inc.: New York, NY, USA, 2014; pp. 3320–3328.
31. Gupta, A.; Ong, Y.; Feng, L. Multifactorial evolution: Toward evolutionary multitasking. *IEEE Trans. Evol. Comput.* **2016**, *20*, 343–357. [CrossRef]
32. Feng, L.; Zhou, L.; Zhong, J.; Gupta, A.; Ong, Y.; Tan, K.; Qin, A.K. Evolutionary multitasking via explicit autoencoding. *IEEE Trans. Cybern.* **2019**, *49*, 3457–3470. [CrossRef]
33. Gong, M.; Tang, Z.; Li, H.; Zhang, J. Evolutionary multitasking with dynamic resource allocating strategy. *IEEE Trans. Evol. Comput.* **2019**, *23*, 858–869. [CrossRef]
34. Frey, B.J.; Dueck, D. Clustering by passing messages between data points. *Science* **2007**, *315*, 972–976. [CrossRef] [PubMed]
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, 7–12 June 2015; pp. 1026–1034.
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. Goodfellow, I.; Warde-Farley, D.; Mirza, M.; Courville, A.; Bengio, Y. Maxout networks. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1319–1327.
38. Lin, M.; Chen, Q.; Yan, S. Network in network. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014; pp. 1–10.
39. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway networks. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–6.
40. Liu, H.; Simonyan, K.; Vinyals, O.; Fernando, C.; Kavukcuoglu, K. Hierarchical representations for efficient architecture search. *arXiv* **2017**, arXiv:1711.00436.

*electronics*

MDPI

*Article*

# Efficiently Mastering the Game of NoGo with Deep Reinforcement Learning Supported by Domain Knowledge

**Yifan Gao** [1,*,†] **and Lezhou Wu** [2,†]

[1] College of Medicine and Biological Information Engineering, Northeastern University, Liaoning 110819, China

[2] College of Information Science and Engineering, Northeastern University, Liaoning 110819, China; 20184005@stu.neu.edu.cn

* Correspondence: yifangao@stumail.neu.edu.cn

† These authors contributed equally to this work.

**Abstract:** Computer games have been regarded as an important field of artificial intelligence (AI) for a long time. The AlphaZero structure has been successful in the game of Go, beating the top professional human players and becoming the baseline method in computer games. However, the AlphaZero training process requires tremendous computing resources, imposing additional difficulties for the AlphaZero-based AI. In this paper, we propose NoGoZero+ to improve the AlphaZero process and apply it to a game similar to Go, NoGo. NoGoZero+ employs several innovative features to improve training speed and performance, and most improvement strategies can be transferred to other nonspecific areas. This paper compares it with the original AlphaZero process, and results show that NoGoZero+ increases the training speed to about six times that of the original AlphaZero process. Moreover, in the experiment, our agent beat the original AlphaZero agent with a score of 81:19 after only being trained by 20,000 self-play games' data (small in quantity compared with 120,000 self-play games' data consumed by the original AlphaZero). The NoGo game program based on NoGoZero+ was the runner-up in the 2020 China Computer Game Championship (CCGC) with limited resources, defeating many AlphaZero-based programs. Our code, pretrained models, and self-play datasets are publicly available. The ultimate goal of this paper is to provide exploratory insights and mature auxiliary tools to enable AI researchers and computer-game communities to study, test, and improve these promising state-of-the-art methods at a much lower cost of computing resources.

**Keywords:** artificial intelligence; deep learning; AlphaZero; NoGo games; reinforcement learning

**Citation:** Gao, Y.; Wu, L. Efficiently Mastering the Game of NoGo with Deep Reinforcement Learning Supported by Domain Knowledge. *Electronics* **2021**, *10*, 1533. https://doi.org/10.3390/electronics10131533

Academic Editor: Amir Mosavi

Received: 31 May 2021
Accepted: 23 June 2021
Published: 24 June 2021

## 1. Introduction

The successive appearance of AlphaGo [1], AlphaGo Zero [2], and AlphaZero [3], achieving remarkable performance in one of the most complex games, Go, demonstrate the capabilities of deep reinforcement learning.

In 2017, Deepmind's AlphaGo Zero showed the possibility for computers to achieve superhuman performance in Go without relying on human knowledge or pre-existing data. Subsequently, AlphaZero made outstanding achievements in chess and shogi. However, a large amount of computational resources was required. Deepmind ran the training progress for Go for several days with 5000 TPUs, while Facebook's ELF OpenGo used 2000 V100 GPUs to achieve the top level of performance [4]. Therefore, it is an important and meaningful research direction to improve AlphaZero with limited computational resources.

This paper introduces several methods to speed up the training process and improve the final performance of the original AlphaGo Zero model (pipeline shown in Figure 1). The reinforced model is called NoGoZero+. Although NoGoZero+ uses some domain-specific features and optimization methods, it still starts from a random policy without using external strategic knowledge or existing data. Additionally, techniques used in

NoGoZero+ can be transferred to other nonspecific domains. By comparing the training process under the same condition in NoGo, the training efficiency of NoGoZero+ was at least six times that of the original AlphaZero.

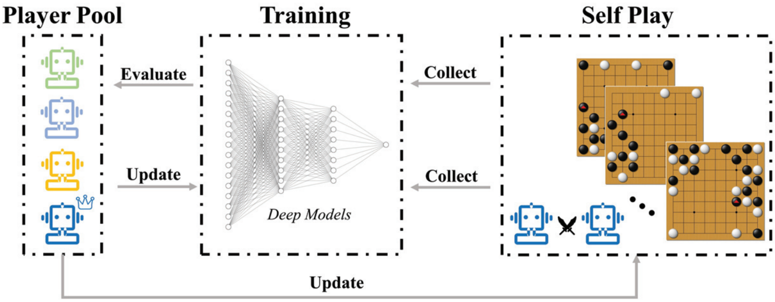


**Figure 1.** AlphaZero's pipeline. Self-play games' data are continuously generated and collected to train deep neural networks. After each round of training, the new model is compared with the previous model. If the new model defeats the previous model, the training process continues. Otherwise, the previous training result is discarded and the previous step is restarted.

NoGo is a new kind of board game that originated in 2005 [5]. Different than traditional Chinese board game Go, it forbids players from capturing stones. Once a player has no choice but to capture the counterpart's stones, the player loses. The formal rules of NoGo are shown below [6]:

- Board size is 9 × 9.
- Black goes first, and both sides take turn in moving in the board. A stone cannot be moved once the location is chosen.
- The goal of both sides is occupying areas instead of capturing counterpart's stones.
- One side loses if it captures the other side's stones or it suicides (deliberately makes its own stone to be captured by their counterpart).

Because of NoGo's novel rules and extremely limited background studies, NoGo does not have a mature strategy. This paper creates a precedent of successfully applying reinforcement learning to NoGo.

The main contributions are summarized as follows:

First, the paper proposes methods that can speed up the training progress and improve the final performance. These methods can be either directly or indirectly applied to similar AlphaZero training processes or general reinforcement-learning processes.

Second, the authors applied NoGoZero+ to the NoGo game and obtained better results than those achieved by the original AlphaZero under the condition of limited resources. This result shows the efficiency gap between AlphaZero's general methods and indicates the existence of better methods under specific conditions.

Third, to help research in this field, we provide the source code used to train the model, some pretrained high-level models, and a comprehensive self-play dataset that contains about 160,000 self-play games (available online: https://github.com/yifangao18/NoGoZero (accessed on 20 June 2021)).

The rest of this paper is organized as follows. In Section 2, we present some closely related works. The paper summarizes the basic architecture in Section 3 and describes the proposed techniques. In Section 4, we introduce experimental settings, criteria, and details. Section 5 reports the result and reviews the performance of NoGoZero+ in the competition. Section 6 discusses the results obtained with our approach. Lastly, we conclude our work and plan future works in Section 7.

## 2. Related Work

In this section, we first go over the AlphaGo family, which first introduced deep reinforcement learning (DRL) to board games with a large searching space, and some famous board game AI based on AlphaGo-like methods. Then, we introduce early work on the game of NoGo. Most of them achieved great success in the early years.

### *2.1. State of the Art in Board Game AI*

The AlphaGo family, including AlphaGo, AlphaGo Zero, and AlphaZero [1–3], have had great success in many complex board games such as Go and chess. Unprecedentedly, AlphaGo family introduced DRL to board games, and DRL-based methods attracted many researchers' attention when AlphaGo beat top human player Lee Sedol. In particular, AlphaGo Zero successfully trains a Go AI from scratch using only the rules of the game because of the successful combination between Monte Carlo tree search (MCTS) [7,8] and deep neural network. Furthermore, the so-called zero-learning method used by AlphaZero is a more general method that can be used both in board games and other, more practical areas.

However, the large amount of computational resources consumed by the training process of AlphaZero and the relatively immature network structures encouraged many researchers to look into improved methods based on AlphaZero. An open-source project called ELF OpenGo [4] reached superhuman level in the game of Go after two weeks of training on 2000 GPUs (a relatively small number compared with the 5000 TPUs used by AlphaZero). KataGo [9], a reimplementation of AlphaGo Zero, improved the learning process in many ways, including using different optimization approaches to have more data about the value in a shorter period of time and using additional training targets to speed up the training process. Leela Zero [10], which is an open-source program trained with GPUs donated by a community of contributors, mastered Go and chess. Morandin et al. [11] proposed a sensible artificial intelligence (SAI) that plays Go to overcome the problem that AIs cannot target their margin of victory, and this is the common problem shared by most of the famous early-game AIs based on AlphaZero. Thus, the SAI successfully overcame the negative consequences: AIs often win by a small margin, cannot be used with komi 6.5, and show lousy play in handicap games.

### *2.2. NoGo AI Research*

NoGo, as opposed to the ancient game of Go, is becoming the new favorite in the game AI community because of its relatively easy rules and lower computational resource requirements. Some early studies on NoGo AI achieved great results. Lee et al. [12] proposed an approach using ontologies, evolutionary computation, fuzzy logic, and fuzzy markup language combined with a genetic-algorithm-based system. Their NoGo AI could analyze the situation of the current board and play the next move to an inferred good-move position. Sun et al. [13] put forward a static-evaluation method to accurately estimate the value of each state of NoGo. Sun et al. [14] successfully used an improved pattern-matching algorithm to find out the best move in the game of NoGo.

As far as we know, there are few studies about the combination of DRL or zero-learning and the game of NoGo. Although former NoGo AI combined with traditional methods achieved relatively good performance, the AlphaGo family showed that there is a great performance gap between traditional methods and the DRL method. As a result, the implementation of DRL and zero-learning method in the game of NoGo is necessary.

## 3. Methods

In this section, we introduce methods that we used to build up NoGoZero+. We first go over the basic architecture of NoGoZero+. Then, we introduce the main novel techniques that we used to improve the training process and the final performance of NoGoZero+.

### *3.1. Basic Architecture*

While NoGoZero+ has various novel details and improvements, it has a similar basic architecture to that of AlphaZero. We basically followed the parameters in [9] because this study provides complete parameter information compared to other AlphaZero implementations.

MCTS is the core searching method of AlphaZero, which is guided by a neural network. NoGoZero+ uses it to play the game against itself to generate training data. It also uses a variant of PUCT [15] to balance exploration and exploitation, and $c_{puct}$ is a constant that determines the importance of both. When $c_{puct}$ is small, MCTS tends to exploit game states with high value. Conversely, when $c_{puct}$ is large, MCTS tends to explore unknown move locations, and the exploration is guided by the policy prior outputs by the neural network. With playouts repeating, the searching process continues, and the search tree grows. The process of playouts starts from the root and goes down the tree, and each node n selects child c with the largest PUCT(c) value:

$$PUCT(c) = V(c) + c_{puct}P(c)\frac{\sqrt{\sum_{c'} N(c')}}{1 + N(c)} \tag{1}$$

where *V(c)* represents the average predicted score of all nodes in *c*'s subtree, *P(c)* represents the policy prior of *c* from the neural network, *N(c)* represents the number of playouts that are used to go through child *c*, and we set $c_{puct} = 1.1$. The Iteration of MCTS is terminated after a certain amount, and generates new policies on the basis of the visit frequency of their child nodes in the tree.

NoGoZero+ adds noise to the policy prior at the root to encourage exploration:

$$P(c) = 0.75P_{raw}(c) + 0.25\eta \tag{2}$$

where $\eta$ is a draw from Dirichlet distribution on legal moves with parameter $\alpha = 0.03 * 9^2/N$, where $N$ is the number of legal moves, and $P_{raw}(c)$ is the policy prior at the root.

The board of NoGo, unlike the traditional 19 × 19 Go board, has a size of 9 × 9, so the $9^2$ part of parameter $\alpha$ corresponds to the NoGo board size. NoGoZero+ also applies a softmax temperature at the root of 1.03 to improve policy convergence stability according to [11].

A convolutional residual network [16] with preactivation is used to guide the search. The residual network has a trunk of *b* residual blocks with *c* channels. However, after various improvements, our network structure was very different compared to that of AlphaZero; see Figure 2. Furthermore, in the primary stage of training, inspired by curriculum learning [17], we designed a method called network curriculum learning to speed up the training progress. NoGoZero+ began with a relatively small residual network and gradually improved the size of the network when it converged under the condition of using the earlier residual network.

### *3.2. Techniques in NoGoZero+*

#### 3.2.1. Global Attention Residual Block

In general, the AlphaZero-based model has a strong ability to capture the local information of the board. However, due to the limited perceptual radius of the classical convolution layer, global strategy prediction remains challenging. Some findings in previous studies on board games, such as the fatal 'ladder failure' that commonly occurs in Go, support this view [4]. Therefore, we began exploring the attention-based structure. As a popular design of computer vision, the attention mechanism can help models to globally pay more attention to important information.

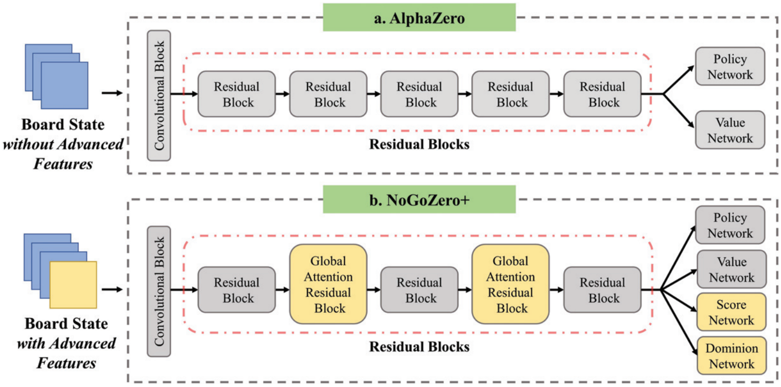


**Figure 2.** Comparison of network architecture of AlphaZero and NoGoZero+ (5 residual blocks).

In this section, we propose the global attention residual block (GARB), a novel attention-based structure based on the global pooling layer, as shown in Figure 3. The global pooling layer was proposed in previous research on the game of Go [9]. The layer enables the convolutional layers to condition in a global context, which can be hard or impossible for convolutional layers with limited perceptual radius. NoGoZero+ replaces parts of the ordinary convolutional layers with the global pooling structure (GPS) to improve the neural network's ability of synthesizing the global context.

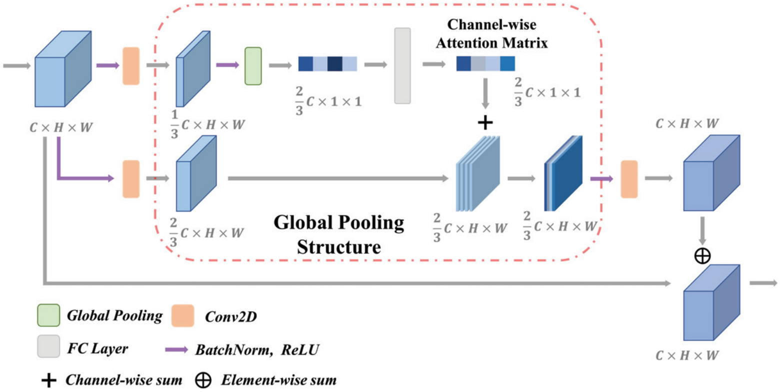

Channel-wise Attention Matrix
$C \times H \times W$
$\frac{1}{3}C \times H \times W$
$\frac{2}{3}C \times 1 \times 1$
$\frac{2}{3}C \times 1 \times 1$
+
$C \times H \times W$
$\frac{2}{3}C \times H \times W$
Global Pooling Structure
$\frac{2}{3}C \times H \times W$
$\frac{2}{3}C \times H \times W$
⊕
$C \times H \times W$
Global Pooling
Conv2D
FC Layer
BatchNorm, ReLU
+ Channel-wise sum
⊕ Element-wise sum

**Figure 3.** Proposed GARB. The structure globally aggregates values of one set of channels to bias another set of channels, potentially providing the final output with information on the global context in NoGoZero+.

Given a set of c channels, the global pooling layer computes the mean of each channel and the maximum of each channel. The whole process outputs a total of 2$c$ values. The global pooling layer is a part of the GPS. Given input $X \in R^{C_1 \times H \times W}$ and $G \in R^{C_2 \times H \times W}$ ($C_1$ and $C_2$ represent the channel, $H$ and $W$ represent the height and width respectively), the GPS includes the following components:

- Batch-normalization layer and rectified linear unit (ReLU) activation function applied to $G$, output shape $C_2 \times H \times W$.
- Global pooling layer applied to $G$, output shape $2C_2$.
- Fully connected (FC) layer to $G$, output shape $C_1$.
- Channelwise sum with $X$, output shape $C_1 \times H \times W$.

Previous studies showed that avoiding dimensionality reduction in the FC layer is crucial for learning channel attention [18]. Therefore, we set the value of $C_1$ to twice that of $C_2$ to avoid performance degradation caused by dimensional changes. GPS was inserted into a standard residual block and eventually constituted GARB. Given input $X_{in} \in R^{C \times H \times W}$ of GARB, two independent convolutional layers decomposed the input into $X$ with channel dimension number $\frac{2C}{3}$, and $G$ with channel dimension number $\frac{C}{3}$ before sending it to GPS.

Moreover, GARB is more focused on extracting long-range dependencies, and it is insufficient for local-information extraction compared to the standard residual block. In order to supplement local information, GARB alternates between two configurations in consecutive residual blocks, as illustrated in Figure 2b.

#### 3.2.2. Multitask Learning

AlphaZero has both a policy head and a value head at the end of its deep neural network. They contribute to the policy network and value network, respectively. The policy head predicts potential good moves, and the value head predicts the final result of the game.The deep neural network outputs policy and value on the basis of the current state of the board. Two parallel networks can be regarded as two training tasks for the neural network.

Such a multitask learning method [19] has had great success in training AlphaZero. NoGoZero+ follows and expands the idea. The paper adds two other output heads, dominion head and score head, which contribute to the two extra tasks (shown in Figure 4a). In NoGo, dominion head predicts the ownership of each location on the board. The ownership indicates the key locations that heavily impact the final result. In both computer games and real world, the final result is suggested both by outputs of a prediction network or noisy 1 and $-1$ (win and loss), and by more details observed during the game. The neural network can have more insight into the cause of the final result, including the true gap between the winner and loser in a playout round.

Our proposed dominion head architecture is shown in Figure 4b. It contains a convolutional block, a global pooling convolutional block, and an output layer. The convolutional block includes a batch-normalization layer, a ReLU activation layer, and a 3 × 3 convolutional layer. The global pooling convolutional block includes the GPS, a batch-normalization layer, a ReLU activation layer, and a 1 × 1 convolutional layer with 2 filters.

We assumed input feature map $x \in R^{C \times H \times W}$, where C, H, and W are channel, height, and width. First, the convolutional block is applied to $x$, output with the same size as that of $x$. Then, the output of the convolutional block is passed through the global pooling convolutional block, and the number of channel dimensions of $x$ is reduced to 2. Lastly, we apply a FC layer with sigmoid activation in the output layer that provides dominion output $y_d \in R^{H \times W}$.

Our proposed score-head architecture is shown in Figure 4c. The network structure of the score head is similar to that of the dominion head, with two differences. The first is that the 1 × 1 convolutional layer of the global pooling convolutional block has only one filter, so the output feature map is 1 in the channel dimension. The second is that the output layer of the score head contains a FC layer with tanh activation, outputting a scalar in the range of $[-1, 1]$. Lastly, we multiply it by the score factor to reflect the score gap between the two players. In NoGo, since most score gaps are within 5 points, the score factor was set to 5, and a larger score gap is regarded as 5 points once appearing. Therefore, output ys is a scalar in the range of $[-5, 5]$.

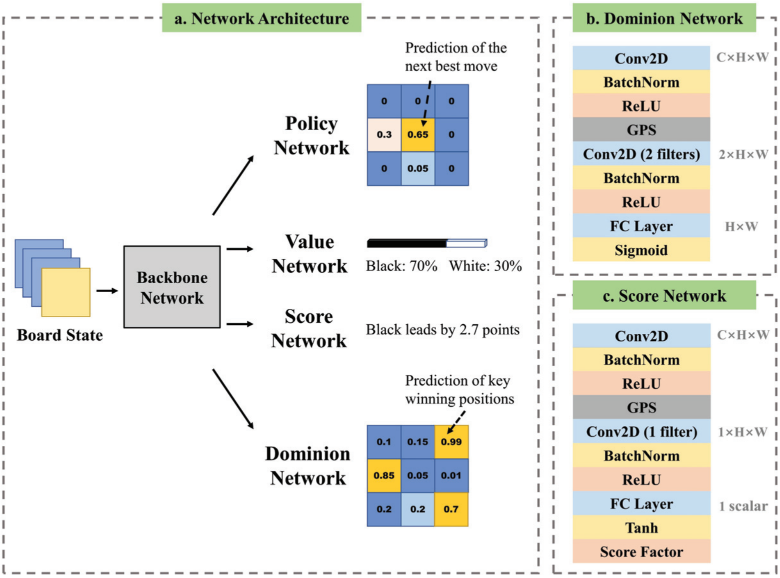


**Figure 4.** (**a**) Multitask learning network structure. (**b**) Illustration of dominion head. (**c**) Illustration of the score head. Except for the policy head and value head, which are also used in AlphaZero, we added score head and dominion head to evaluate the gap of the performance of two players and the key winning position, respectively. The two additional heads can effectively help the NoGoZero+ agent have insight into the game state and make full use of self-play data because of the more detailed information.

### 3.2.3. Network Curriculum Learning

Curriculum learning is an important method in reinforcement learning [20,21]. To prevent the agent from being stuck at the early stage of training because of the hard initial task or having dissatisfactory performance at the end of training because of oversimplified tasks, we carried out curriculum learning by imitating the learning process of humans and animals and gradually improve the model trough progressive samples and knowledge.

In this paper, the idea of curriculum learning was extended to the neural network and is called 'network curriculum learning'. A ahallower and narrower structure leads to a network with less complexity, but it converges more quickly. While the deeper and wider network with more complexity is usually harder to be trained, network curriculum learning tries to balance complexity and convergence rate. During NoGoZero+ training, the process starts with a simpler network with fewer parameters. Self-play and training data are generated by the shallow network, and used to train a deeper and more complex network (and the simpler network itself), while the deeper network itself does not generate new data. As soon as the simpler network meets its bottleneck, which means that this network has converged, the whole training process is transferred to a larger network. The process is repeated until the size and the performance of the network satisfy our needs.

Figure 5 shows the course of network curriculum learning. The whole process of training the NoGoZero+ agent can be separated into three stages, and the number of residual blocks (b in Figure 5) grew from 5 to 20, while the number of channels (c in Figure 5) of each residual block grew from 32 to 128. With the help of network curriculum learning, the agent avoids the stagnation of training in the early stage and saves much training time.

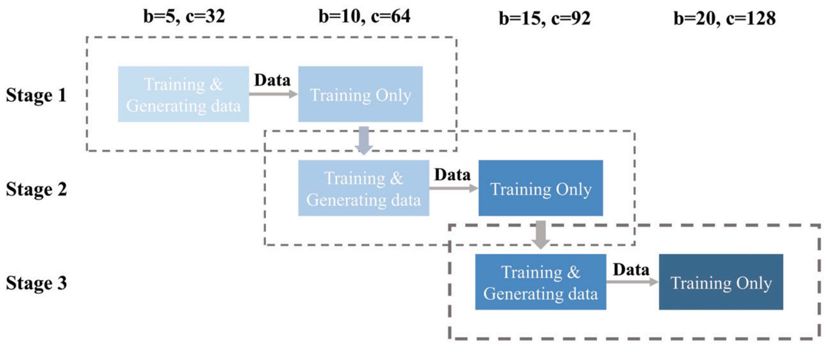


**Figure 5.** Process of network curriculum learning. Data generated by the shallower network are used to train both the shallow network itself and a deeper network. The whole process contains three stages, and the network with b = 20, c = 128 was extracted as the final network that is used in the agent.

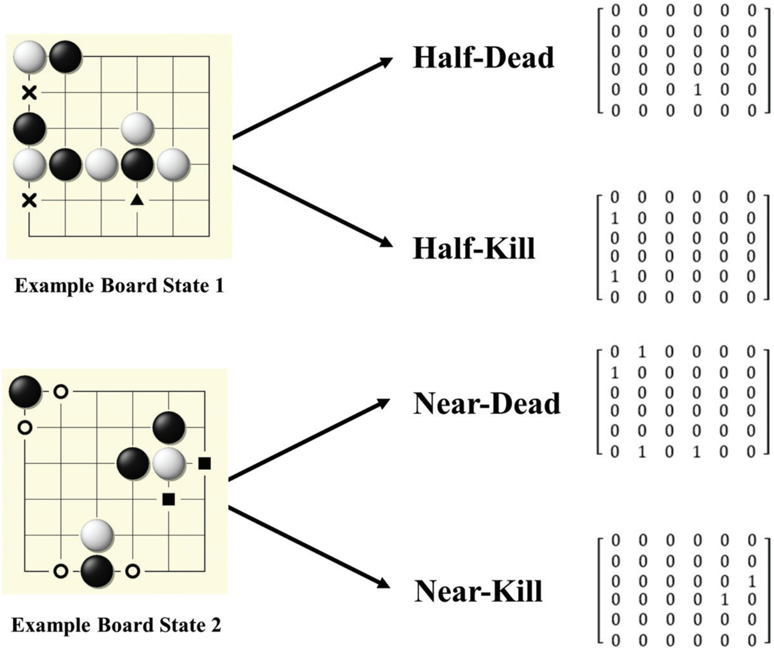
### 3.2.4. Advanced Specific Features

AlphaZero provides a universal method to train game agents. The method performs perfectly on chess, shogi, and Go. However, a sea of computational resources that make the universal method work. With limited resources, it is important for us to add some advanced specific features to speed up the training process and make full use of the obtained data from every game. Like other machine-learning methods [22], reinforcement learning also attaches much importance to well-designed specific features, which can make a great difference to the speed of training process, as well as the final performance of agents.

Similar to imitation learning, some domain-specific professional knowledge guides the deep model to converge faster during the initial stage. Besides current and historical steps, which are also the features used by AlphaZero, NoGoZero+ adds another kind of advanced feature, liberty, which is also called 'qi'. Four input layers are added on the basis of the input layers of the network to show the features of liberty:

- Liberty locations belonging to our stones that have only one liberty.
- Liberty locations belonging to our stones that have two liberties.
- Liberty locations belonging to the counterpart's stones that have only one liberty.
- Liberty locations belonging to the counterpart's stones that have two liberties.

The four above features are (1) 'half-dead', (2) 'near-dead', (3) 'half-kill', and (4) 'near-kill', respectively. One-hot encoding is used to represent the four advanced features. The corresponding location in the input layer is set to 1 once a specific feature appears. The corresponding features are schematically shown in Figure 6.



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

**Figure 6.** Demonstration of advanced features. Location of existing features is marked with one-hot encoding in the corresponding input layer. Four additional layers are added to the input layer set of the original AlphaZero and provide the NoGoZero+ agent with additional advanced information about the board state. Results in the next section show that such modification can obviously improve training speed and final performance.

## 4. Experiments

In the two following sections, we use some control tests to verify the improvements our techniques bring to the basic AlphaZero method, and use ablation experiments to examine the contributions of each approach used in NoGoZero+. In this section, we introduce the experimental settings, evaluation criteria, and training parameters to detail the background and process of our experiments.

### *4.1. Experimental Settings*

To look into the behavior of NoGoZero+ and demonstrate the effects of the unique techniques introduced in the paper, two experiments were conducted.

First, we designed an experiment to show the significant speedups attributing to adding extra training methods to the original AlphaZero theory by comparing three models' training processes on the NoGo:

- Training process of the original AlphaZero.
- Training process of AlphaZero with network-curriculum-learning technique.
- Training process of NoGoZero+.

Second, to explore the effects of every single technique, an ablation experiment was conducted. To more clearly demonstrate the contributions of each technique to the final result, all agents in the ablation experiment did not use the network-curriculum-learning method.

### *4.2. Evaluation Criteria*

The Elo model [23] is an effective way to measure the level of game agents, widely used to evaluate the performance of agents in, for example, chess, Go, basketball, and football. Assuming that the Elo ratings of Agents $A$ and $B$ are $R_A$ and $R_B$, respectively, the expected winning probability of $A$ versus $B$ is:

$$E_A = \frac{1}{1 + 10^{\frac{R_B - R_A}{400}}} \tag{3}$$

When Player $A$'s true score $S_A$ is different from its expected winning probability (1 for win, 0 for loss, 0.5 for tie. There is no tie in NoGo.), its Elo rating should be adjusted as

$$R_A^{'} = R_A + \alpha(S_A - E_A) \tag{4}$$

where $R'_A$ is the Elo rating of Player $A$ after adjustment, and $\alpha$ is a weight that is often set to 16 in master tournaments ($\alpha$ = 16 is also used in our experiment).

### *4.3. Training Parameters and Details*

We set stochastic gradient descent (SGD) as the optimizer with a learning rate equal to 0.001 and momentum equal to 0.8. Our experiments were performed on an NVIDIA Tesla V100 GPU (with 32 GB GPU memory). The deep-learning models were implemented using Pytorch. In training, the data-augmentation technique that we used includes all eight reflections and rotations for each position. The entire training process took about 100 h, and most of the resources were consumed in self-play. Multithreading self-play is recommended to make full use of GPU and CPU resources.

Considering the four output heads of the improved network structure, the loss function is the sum of the five following kinds of losses.

- Policy Loss

  $$-\sum_m \pi(m) \log\left(\hat{\pi}(m)\right) \tag{5}$$

  where $m$ is the set of the rest legal moves, $\pi$ is the target policy derived from the playouts of the MCTS search, and $\hat{\pi}$ is the neural network's prediction of policy $\pi$.

- Value Loss

  $$-c_g \sum_r z(r) \log\left(\hat{z}(r)\right) \tag{6}$$

  where $r$ is the final result for the current player (($r \in \{win,\ lost\}$), $z$ is a one-hot encoding function of $r$, $\hat{z}$ is the neural network's prediction of the final result. $c_g$ = 1.5 is a scaling constant.
- Dominion Loss

  $$-c_d \sum_m \gamma(m) \log\left(\hat{\gamma}(m)\right) + (1 - \gamma(m)) \log\left(1 - \hat{\gamma}(m)\right) \tag{7}$$

  where $m$ is the set of the moves in the board, $\gamma$ is the final result for the current player ($r \in \{dominion,\ not\ dominion\}$), $\hat{z}$ is the neural network's prediction of $\gamma$. $c_d$ = 0.5 is a scaling constant.
- Score Loss

  $$c_s(\hat{v} - v)^2 \tag{8}$$

  where $v$ is the scalar of score difference, $\hat{v}$ is the prediction of the final result. $c_s$ = 0.25 is a scaling constant.
- L2 Penalty

  $$c_{L2}\|\theta\|^2 \tag{9}$$

  where $c_{L2}$ = $10^{-5}$, so as to prevent the network from overfitting due to the relatively deep neural network structure.

## 5. Results

In this section, the experiment results are shown and discussed. Then, some interesting playouts are displayed and explained to demonstrate the intelligence of NoGoZero+. Lastly, we introduce the performance of NoGoZero+ in the competition and summarize the possible reasons for not winning first place.

### *5.1. Experiment Results*

The results of the first experiment are shown in Table 1. AlphaZero's Elo rating reached 2500 after 120,000 self-play games, while it only took NoGoZero+ 20,000 self-play games to reach an Elo rating of 2750, and NoGoZero+ defeated original AlphaZero with 81 wins and 19 losses in 100 playouts. The AlphaZero agent, with the help of the network-curriculum-learning method, had a similar learning curve to that of NoGoZero+. However, an obvious gap of Elo rating between the AlphaZero agent with network curriculum learning and complete NoGoZero+ indicated that other techniques that we use make a great difference in the behavior of the agent.

**Table 1.** Comparison of Elo ratings among original AlphaZero, AlphaZero with only network curriculum learning, and complete NoGoZero+. NCL, network curriculum learning. NoGoZero+ had advantages over the original AlphaZero in final performance (Elo rating) and training speed. Moreover, the obvious gap between complete NoGoZero+ and AlphaZero with only network curriculum learning indicated the supplementary power of other techniques. Blocks represent the number of residual blocks in the network.

| Method | Blocks | NCL | Games | Elo Score |
|---|---|---|---|---|
| AlphaZero | 20b | | 120k | 2500 |
| AlphaZero | 5b | ✓ | 3k | 1800 |
| AlphaZero | 10b | ✓ | 10k | 2250 |
| AlphaZero | 20b | ✓ | 20k | 2350 |
| NoGoZero+ | 5b | ✓ | 3k | 2300 |
| NoGoZero+ | 10b | ✓ | 10k | 2625 |
| NoGoZero+ | 20b | ✓ | 20k | 2750 |

The results of the second experiment (ablation experiments) are shown in Table 2. Comparing the final behavior of the agents with different parts weeded out, GARB had a relatively larger influence on the final performance (the Elo rating of which dropped from 2300 to 2045 without it), while multitask learning and advanced features closely impacted the performance of the agent. Adding advanced features was less influential because the main aim of adding advanced features is to help the agent make full use of detailed information obtained from one single playout to speed up the training process with relatively limited resources. Overall, every technique used in the training process of NoGoZero+ had a positive effect, which can effectively help to improve the Elo rating of the agent.

### *5.2. Strategies Learnt by the Agent*

Unlike Go, which originated thousands of years ago and has been researched for quite a long time, NoGo is a new board game with few mature strategies. However, with the help of techniques shown in the previous sections, the NoGoZero+ agent developed a series of inspirational strategies. The paper demonstrates and discusses some distinct, impressive strategies shown by the NoGoZero+ agent in self-play games.

**Table 2.** Comparison of Elo ratings among original AlphaZero, AlphaZero with only network curriculum learning, AlphaZero using some of the techniques in this paper, and complete NoGoZero+. ASF, advanced specific features; MTL, multitask learning network structure; 5b, number of residual blocks in the network.

| AlphaZero (5b) | GARB | MTL | ASF | Games | Elo Score |
|---|---|---|---|---|---|
| ✓ | | | | 3k | 1800 |
| ✓ | ✓ | ✓ | | 3k | 2174 ± 47.62 |
| ✓ | ✓ | | ✓ | 3k | 2134 ± 52.24 |
| ✓ | | ✓ | ✓ | 3k | 2045 ± 87.39 |
| ✓ | ✓ | ✓ | ✓ | 3k | 2300 |

#### 5.2.1. Triangle Strategy

A triangle layout and a prismatic layout are two kinds of favorable layouts to players because the central location of such layouts cannot be occupied by a counterpart's stones. As a result, such layouts can greatly help at the end of the game when there are few locations to place a stone. At the beginning of a game, the agent tries to build as many triangle and prismatic layouts as possible, if not disturbed by the counterpart player (see Figure 7). Moreover, the agent sometimes manages to prevent its counterpart from successfully building a triangle or a prismatic layout after weighing advantages and disadvantages. Such behavior is called the 'triangle strategy'.

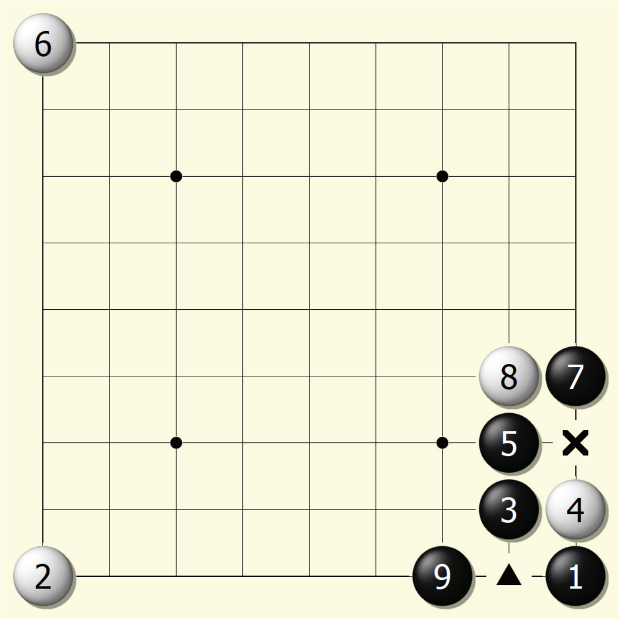


**Figure 7.** Demonstration of 'triangle strategy'. Black successfully built a triangle structure in the location marked by △, and white successfully prevented black from building triangle structure in Step 4 because neither black nor white could occupy the location marked by ×. Moreover, the white stone placed in Step 8 also tried to destroy a potential triangle structure that may have contained the black stones placed in Steps 5 and 7.

#### 5.2.2. Predictive Strategy

With the progress of the NoGoZero+ agent, it becomes harder for white to win (the average win rate is 4.5) because black learns to make full use of the sente advantage.

A typical situation is: white tries hard to prevent black from using the triangle strategy and fails because black takes the lead in the order (Figure 8).

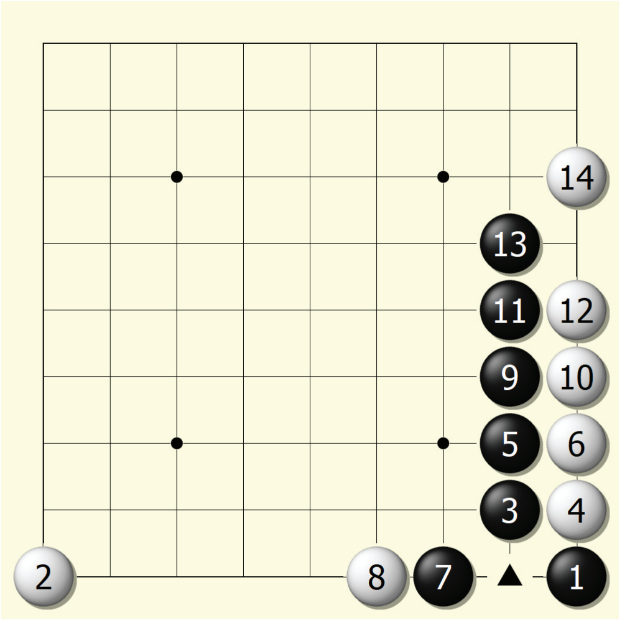


**Figure 8.** A typical situation where the white falls behind. Starting from Step 4, white tried hard to prevent black from using the triangle strategy, but black always took the lead. Although white made a wise choice in Step 14 and terminated the 'chasing game', black successfully built a triangle structure in Step 7. The whole process indicates that black has an obvious advantage over white when the NoGoZero+ agent is well-trained.

Since NoGo is considered to be a game where the black player is more likely to win, the white player needs to adopt a more aggressive strategy to win the game. We looked into games in which white won, and found that white seemed to gain foresight under certain conditions.

Figure 9 shows a part of a game where the white player won. Instead of chasing black's steps and trying hard to stop black in a relatively small area, the white player controlled the overall situation and successfully disturbed black's deployment on a larger scale.

Such a decision shows a sense of prediction, so the behavior is called 'predictive strategy'. The sense of the overall situation and the prediction should be attributed to the GARB, which gives the agent a high level of understanding of the whole situation, and multitask learning makes the agent capable of predicting the behavior of its counterpart.

Although the white player did not fully understand the predictive strategy in the experiment, it is impressive that the agent developed such a high-level tactic. Hopefully, after more self-play games, the agent can master predictive strategy or even other high-level strategies.

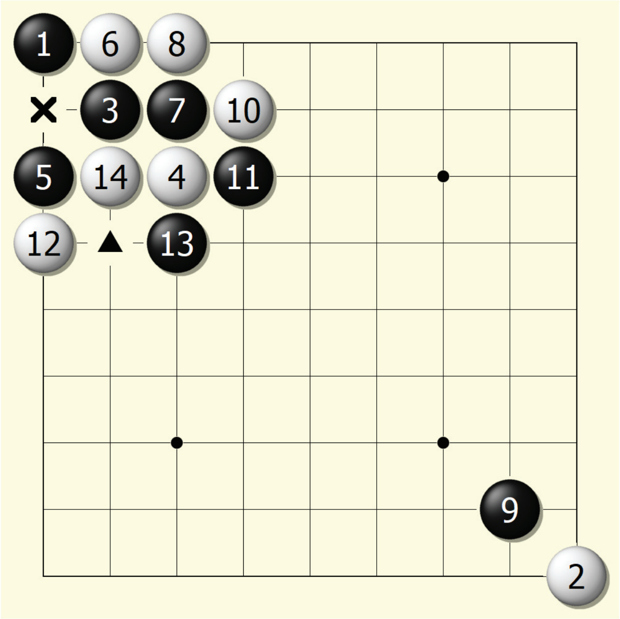


**Figure 9.** Demonstration of predictive strategy. When black tried to build a triangle structure in Step 3, instead of chasing black's step like the behavior shown in Figure 8, white took Step 4 and built a wider encirclement, thus rendering the triangle structure built by black (marked by ×) useless.

### *5.3. NoGoZero+ in CCGC*

The China Computer Game Championship (CCGC) is the largest computer-game competition in China, held annually since 2006. In 2020, a total of 18 teams entered the national finals. Our team competed for the first time and unfortunately lost in the finals to the KnighTeam-NG (KNG) program developed by the Chongqing University of Technology. KNG has won consecutive championships in NoGo since 2014. As the code is not open-source, we cannot know what kind of technology KNG uses.

One lesson is that we unreasonably allocated the game time. NoGoZero+ spent much time in the first half of the game, and when the second half of the time was exhausted, some strange and unreasonable moves appeared before it lastly lost the game. In addition, our equipment for participating in the competition was not good enough. For each move, NoGoZero+ could only perform 1400–1800 MCTS on average, which led to worse model performance.

## 6. Discussion

Although the main battlefield of NoGoZero+ is still the game of NoGo, the techniques used by NoGoZero+ can be extended to other nonspecific areas to reduce resource consumption caused by large search spaces. First, GARB is an efficient attention mechanism that helps improve the network's ability to perceive global information. Second, the success of the multitask learning network structure proved that adding more output heads to neural networks can efficiently help the training process. Third, the network-curriculum-learning technique can help to balance the complexity and the rate of convergence of a deep neural network. Fourth, adding moderate domain-specific knowledge to the training process of neural networks can make full use of training data and reduce the training period, especially when computational resources are limited.

The game of NoGo is becoming a useful tool for many researchers who are interested in game AI but do not want to spend too much time learning the complex game rules to carry out studies. On the basis of the NoGoZero+ result, they can quickly develop a basic structure of

NoGo game AI and save time for further study. As a result, with the help of techniques used by NoGoZero+, the game AI community benefits from the lower research thresholds.

Moreover, AlphaZero-based methods mainly use convolutional neural networks (CNN) to capture board information, just like CNN captures information from pictures in computer-vision tasks. This means that many tricks used in computer vision can be transformed into AI training processes. The attention model, which is commonly used to select the most crucial information from pictures during computer-vision tasks, can be combined with the original CNN structure to improve efficiency in the usage of computational resources, and further enhance the final performance of the AI. This method is reasonable and needs further research because some positions are also more important than others are in board games.

## 7. Conclusions and Future Work

This paper presented NoGoZero+, which efficiently masters the game of NoGo on the basis of the improved AlphaZero algorithm. By using several techniques, the improvement is nearly cost-free. The methods enable a unified reinforcement-learning-based system to be trained from scratch to being a powerful agent in a few days. The techniques used in NoGoZero+ can easily be extended to other areas to reduce computational-resource consumption and improve training efficiency. Experiments showed that NoGoZero+ had six times better training speed and better performance than those of the original AlphaZero. This study highlights the tremendous potential of reducing computing resources in board-game AI.

Although the experiment results of this study are encouraging, there are still some limitations. For example, we were only awarded second place in the CCGC. The future of this study includes exploring more powerful neural-network architectures and more efficient sample-utilization methods to further improve performance. Our ultimate goal is to provide insights and additional tools for the community to explore large-scale deep-learning methods of computer games. Our code and data are public to help researchers in the game AI community.

**Author Contributions:** Conceptualization, Y.G. and L.W.; methodology, L.W. and Y.G.; validation, L.W. and Y.G.; resources, Y.G. and L.W.; data curation, Y.G. and L.W.; writing—original-draft preparation, Y.G. and L.W.; writing—review and editing, L.W. and Y.G.; supervision—L.W. and Y.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** The authors thank Jiao Wang from the College of Information Science and Engineering, Northeastern University, China for their support in preparing this study.

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

## References

1. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
2. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of Go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef] [PubMed]
3. Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **2018**, *362*, 1140–1144. [CrossRef] [PubMed]
4. Tian, Y.; Ma, J.; Gong, Q.; Sengupta, S.; Chen, Z.; Pinkerton, J.; Zitnick, L. Elf OpenGo: An analysis and open reimplementation of AlphaZero. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6244–6253.
5. Moore, J. NoGo History and Competitions. Available online: http://webdocs.cs.ualberta.ca/~mmueller/nogo/history.html (accessed on 24 June 2021).
6. CCGC Organizing Committee. University Computer Games Championship & National Computer Games Tournament. Available online: http://computergames.caai.cn/jsgz05.html (accessed on 24 June 2021).

7. Coulom, R. Efficient selectivity and backup operators in Monte-Carlo tree search. In Proceedings of the International Conference on Computers and Games, Turin, Italy, 29–31 May 2006; pp. 72–83.
8. Kocsis, L.; Szepesvári, C. Bandit based Monte-Carlo planning. In Proceedings of the European Conference on Machine Learning (ECML), Berlin, Germany, 18–22 September 2006; pp. 282–293.
9. Wu, D.J. Accelerating self-play learning in Go. *arXiv* **2019**, arXiv:1902.10565.
10. Pascutto, G.-C. Leela Zero. Available online: https://github.com/leela-zero/leela-zero (accessed on 24 June 2021).
11. Morandin, F.; Amato, G.; Gini, R.; Metta, C.; Parton, M.; Pascutto, G.C. SAI, a Sensible Artificial Intelligence that plays Go. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8. [CrossRef]
12. Lee, C.S.; Wang, M.H.; Chen, Y.J.; Hagras, H.; Wu, M.J.; Teytaud, O. Genetic fuzzy markup language for game of NoGo. *Knowl. Based Syst.* **2012**, *34*, 64–80. [CrossRef]
13. Sun, Y.; Rao, G.; Sun, H.; Wei, Y. Research on static evaluation method for computer game of NoGo. In Proceedings of the 26th Chinese Control and Decision Conference, Changsha, China, 31 May–2 June 2014; pp. 3455–3459. [CrossRef]
14. Sun, Y.; Wang, Y.; Li, F. Pattern matching and Monte-Carlo simulation mechanism for the game of NoGo. In Proceedings of the IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, Hangzhou, China, 30 October 2012; Volume 1, pp. 61–64. [CrossRef]
15. Rosin, C.D. Multi-armed bandits with episode context. *Ann. Math. Artif. Intell.* **2011**, *61*, 203–230. [CrossRef]
16. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
17. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum learning. In Proceedings of the International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 41–48.
18. 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–18 June 2020.
19. Caruana, R. Multitask learning. *Mach. Learn.* **1997**, *28*, 41–75. [CrossRef]
20. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
21. Szepesvári, C. Algorithms for reinforcement learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2010**, *4*, 1–103. [CrossRef]
22. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Prentice Hall: Hoboken, NJ, USA, 2002.
23. Elo, A. *The Rating of Chess Players, Past and Present*; Arco Publishing: London, UK, 1978.

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland
Tel. +41 61 683 77 34
Fax +41 61 302 89 18
www.mdpi.com

*Electronics* Editorial Office
E-mail: electronics@mdpi.com
www.mdpi.com/journal/electronics

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland

Tel: +41 61 683 77 34

www.mdpi.com

ISBN 978-3-0365-7039-6