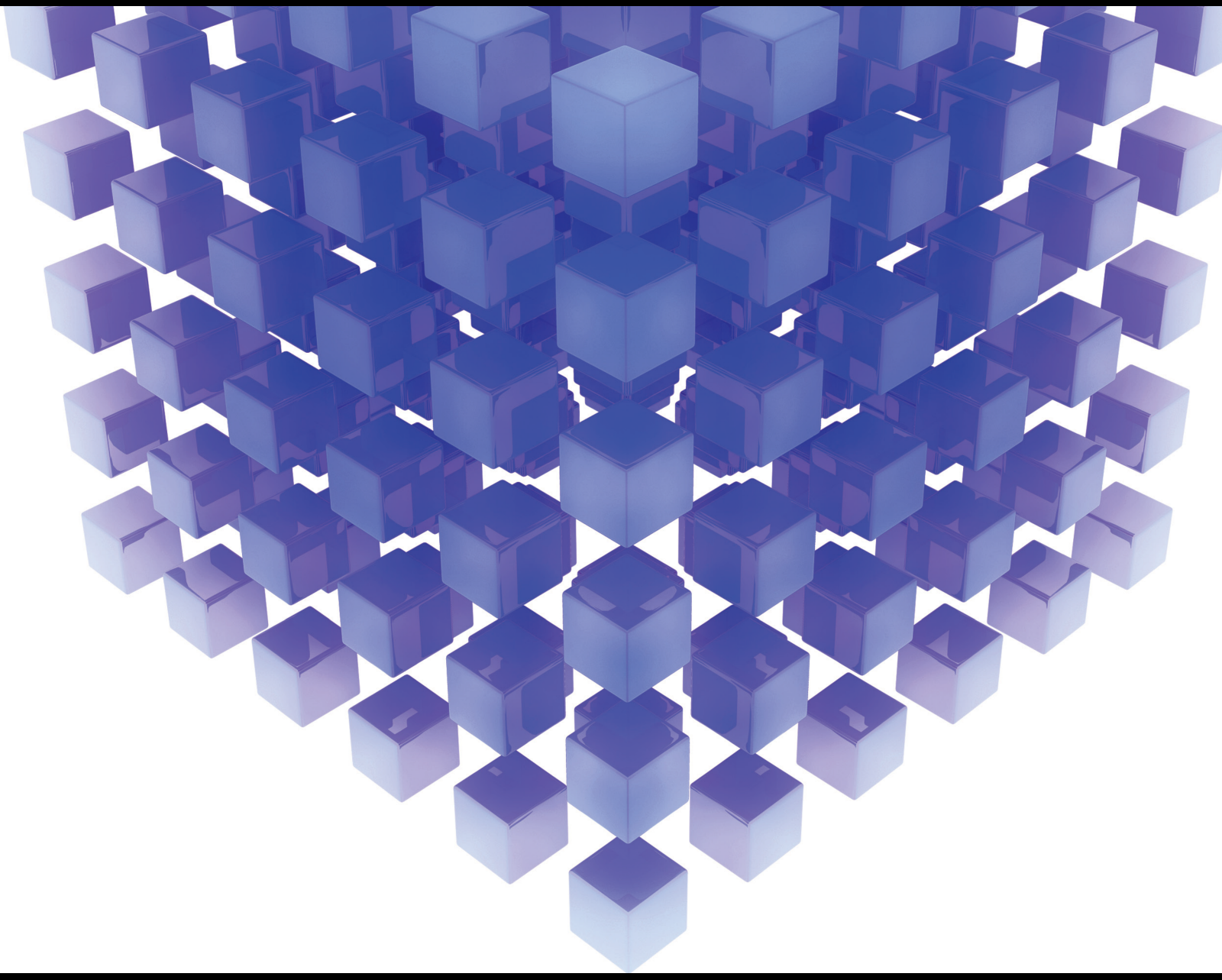
Mathematical Problems in Engineering

Machine Learning, Deep Learning, and Optimization Techniques for Transportation

Lead Guest Editor: Chi-Hua Chen
Guest Editors: Feng-Jang Hwang, Chunjia Han, and Fangying Song



Machine Learning, Deep Learning, and Optimization Techniques for Transportation

Mathematical Problems in Engineering

Machine Learning, Deep Learning, and Optimization Techniques for Transportation

Lead Guest Editor: Chi-Hua Chen
Guest Editors: Feng-Jang Hwang, Chunjia Han, and Fangying Song

Contents

Contents

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 4813183, 10 pages
https://doi.org/10.1155/2020/4813183

Research Article

Lightweight Ship Detection Methods Based on YOLOv3 and DenseNet

Zhelin Li, Lining Zhao, Xu Han, and Mingyang Pan

Dalian Maritime University, Dalian 116026, China

Correspondence should be addressed to Mingyang Pan; panmingyang@dlmu.edu.cn

Received 20 April 2020; Revised 10 September 2020; Accepted 11 September 2020; Published 28 September 2020

Academic Editor: Feng-Jang Hwang

Ship detection is one of the most important research contents of ship intelligent navigation and monitoring. As a supplement to classical navigational equipment such as radar and the Automatic Identification System (AIS), target detection based on computer vision and deep learning has become a new important method. A target detector called YOLOv3 has the advantages of detection speed and accuracy and meets the real-time requirements for ship detection. However, YOLOv3 has a large number of backbone network parameters and requires high hardware performance, which is not conducive to the popularization of applications. On the basis of YOLOv3, this paper proposes a lightweight ship detection model (LSDM) in which the backbone network is improved by using dense connection inspired from DenseNet, and the feature pyramid networks are improved by using spatial separation convolution to replace normal convolution. The two improvements reduce parameters and optimize the network structure greatly. The experimental results show that, with only one-third of parameters of YOLOv3, the LSDM has higher accuracy and speed for ship detection. In addition, the LSDM is simplified further by reducing the number of densely connected units to form a model called LSDM-tiny. The experimental results show that, LSDM-tiny has similar detection speed with YOLOv3-tiny, but has a lot higher accuracy.

1. Introduction

In the recent years, object detection technologies based on deep learning have received more and more attention in the areas of ship intelligent navigation and ship monitoring [1, 2]. The rapid perception of the navigational environment is the prerequisite for ships to sail safely or for shore base to monitor ships real-time [3]. The perception data collected by radar and the AIS play an important role in these application areas [4]. However, how to detect the ship targets by cameras visually and automatically has become a new challenge for realizing autonomous navigation or intelligent supervision.

Normally, there are two main types of object detection algorithms based on deep learning. One is a two-stage detection algorithm, such as Fast R-CNN [5] and its improved versions Faster R-CNN [6] and Mask R-CNN [7]. Two-stage algorithms first select candidate targets through the regional proposal networks (RPN) and, then, complete the prediction of the target location and category through the detection network. The other type is the one-stage detection algorithm, such as SSD [8], YOLO [9], and RetinaNet [10]. One-stage algorithm eliminates the step of region proposal and directly returns the target position and category. All detection algorithms are generally trained on large public data sets to pursue high accuracy; however, they also require high hardware performance to be trained and executed.

Unlike ordinary detection models, lightweight detection models with smaller parameters aim to run on mobile devices or computers with weak computing capabilities. Iandola et al. proposed a lightweight model called SqueezeNet, in which the parameters are compressed to one-fifth of AlexNet by using a small convolution kernel and reducing the number of input and output channels of the convolution layer [11]. Huang et al. proposed a densely connected network called DenseNet, in which feature reuse was used to reduce the parameter of deep networks [12]. Howard et al. built a lightweight deep neural network called MobileNets through deep separable convolutions [13]. Zhang et al.

proposed an efficient convolutional neural network called ShuffleNet, in which the computational cost was greatly reduced through pointwise group convolution and channel shuffling while ensuring accuracy [14].

For ship detection, in addition to accuracy, improving the detection speed is also important to adapt the model to existing hardware condition. As the YOLOv3 has a relatively balanced performance in detection accuracy and speed [15] and the DenseNet has obvious effect on reducing parameters, this paper focused on researching lightweight ship detection models by combining the YOLOv3 and DenseNet and provided a new lightweight detector for high-accuracy ship detection. The contribution of this paper includes the following:

(1) We propose a lightweight ship detection model called the LSDM, with one-third parameters of the YOLOv3 network, and higher average accuracy of 94% for ship detection

(2) We propose a simpler version of the LSDM called LSDM-tiny, with one-eighth parameters of the YOLOv3 network, double detection speed, and average accuracy of 93.5% for ship detection

The rest of the paper is organized as follows. Section 2 outlines the related works of ship detection, Section 3 introduces the network details of the LSDM and LSDM-tiny based on YOLOv3 and DenseNet. Section 4 gives the experimental results and comparative analysis. Finally, Section 5 gives the conclusions and future work.

2. Related Works

For image detection, many studies have made improvements on the basic detection model. Fang et al. used DCGAN to generate samples and training in the image recognition model which based by the CNN to improve the accuracy of image recognition [16]. Meng et al. proposed an approach based on the Faster R-CNN which corresponds multiple steganographic algorithms to complex texture objects was presented for hiding secret messages [17]. Cui et al. proposed an effective approach to automatically identify photographic images and computer-generated graphics based on deep convolutional neural networks (DCNNs) by deepening the network structure [18].

For ship detection, there are two types of images to be utilized, radar image and visible image. Generally, the radar image covers a wider range, and the visible image provides more detailed information. Dong et al. improved the R-CNN and proposed a multiangle box-based rotation insensitive object detection structure for detecting VHR (Very-High-Resolution) ship images [19]. Yang et al. proposed a detection model based on the multitask rotational region convolutional neural network to detect dense ships and predict the direction of ship navigation [20]. Fan et al. proposed a ship detection method for PolSAR (Polarization Synthetic Aperture Radar) images based on modified the Faster R-CNN which is still difficult to detect small targets near the shore [21]. Zhang et al. optimized the backbone network of the Faster R-CNN by using SVM to divide the area of interest and improved effectively the ship detection effect from SAR images [22]. Jiao et al. proposed a Faster R-CNN detection framework based on densely connected multiscale neural networks for detecting SAR ship targets in multiscale and multiscenarios [23]. Kim et al. proposed a hybrid method in which Faster R-CNN was used to detect ships in each image, and the detected ships are, then, gathered over time to compute probabilities for Bayesian fusion to determine the classification of ships [24]. The abovementioned methods are two-stage algorithms which have effectively improved detection effect, but need a long detection time to extract proposal regions through the selective search algorithm or RPN, so they are difficult to implement real-time detection.

There are also many ship detection methods based on one-stage detection algorithms. An et al. proposed an improved RBox-based target detection framework to improve detection accuracy and recall [25]. Liu et al. proposed an improved YOLOv3 algorithm based on the Darknet to realize the detection and tracking of ships in monitored water areas [26]. Zhang et al. proposed a ship target tracking algorithm based on the YOLO method in which the characteristics of HOG and LBP were combined to solve the problem of missing or inaccurate positioning [27]. Wang et al. used SSD to perform transfer learning on Sentinel-1 SAR ship images, improving the detection accuracy and overall performance [28]. Wang et al. used RetinaNet to detect multiscale Gaofen-3 ship images and obtained higher detection accuracy [29]. Chang et al. proposed a YOLOv2 detection structure with reduced number of layers for detecting SAR ship images, which accelerated the inference process while maintaining similar detection effects [30]. Zhang and Zhang drew on YOLO's regression ideas and proposed a detection structure composed of a backbone convolutional network and a detection convolutional neural network, which has a faster detection speed [31]. Generally, one-stage algorithms have higher detection speed but worse accuracy than two-stage algorithms. The abovementioned methods mainly improved the detection accuracy based on one-stage detection algorithms. However, their backbone networks still have a large number of parameters, which not only bring the risk of overfitting and a big model size but also constrain them to archive higher detection speed.

To implement real-time ship detection, Qi et al. proposed an improved Faster R-CNN algorithm by scene reduction technology to reduce the target scale during searching [32]. Lin et al. proposed a new network for ship detection in SAR Images based on the Faster R-CNN, which improved the detection performance and execution speed through squeeze and excitation mechanisms [33]. Zhang et al. proposed a ship detection model called CCNet which uses a cascaded CNN model with REM and RDM, which has five times less computation than other algorithms [34]. The abovementioned methods tried to improve the detection performance and speed based on two-stage algorithms by reducing the weight of the object detection proposal region. However, compared to one-stage end-to-end models, the candidate areas selection step of the methods is still an unignorable factor which would slow down the detection

speed obviously. Zhang et al. proposed a high-speed SAR ship detection model inspired from the experiments of MobileNet, YOLO, SSD, and DenseNet, in which deep separable convolutions instead of ordinary convolutions were used to reduce parameters and improve detection speed significantly; however, it leads to part loss of accuracy [35].

So, with review and analysis of the related works of ship detection, this paper aims to study lightweight ship detection models base on one-stage algorithm, which would keep the accuracy as is possible while reducing the number of parameters and increasing the detection speed.

3. Lightweight Ship Detection Methods

3.1. YOLOv3. YOLOv3 is an end-to-end object detection model, and its network structure includes a backbone network and a detection network [15], as shown in Figure 1. In the backbone network, an input image is scaled to the size of 416* 416 without changing the aspect ratio, and down sampled 5 times to extract feature maps. Then, the feature maps with the size of 13*13, 26*26, 52*52 are output, respectively, to three branches of the detection network to form a feature pyramid structure, in which the feature maps in a lower branch are concatenated with the ones in its next branch by up sampling. Finally, the outputs of the feature pyramid network are sent to a regression section to carry out bounding box and category prediction.

The original backbone network of YOLOv3 is Darknet-53. Darknet-53 includes 52 fully convolution layers, in which 46 layers are divided into 23 residual units with 5 different sizes [15]. The residual units are designed to avoid the vanishing-gradient problem inspired from the Resnet [36]. The Darknet-53 is a complex network, and its 40549216 parameters provide a guarantee for the detection accuracy. However, for the object detection with single category such as ship, the excessively huge parameters would bring overfitting risk and slow down the detection speed.

YOLOv3-tiny is a simplified version of YOLOv3, its backbone network only includes 7 convolutional layers and 6 max-pooling layers, and its feature pyramid network is also simplified by removing the maximum-scale prediction branch and reducing the number of convolutional layers in the other two branches. So, YOLOv3-tiny has faster detection speed than YOLOv3 due to its shallow and simple network structure; however, its detection accuracy is lower obviously than YOLOv3.

Therefore, for fast ship detection, it is important to maintain the depth of network for capturing enough features to ensure detection accuracy while reducing network parameters to speed up. In addition, the ship objects are relatively small in the images, when detecting them by Darknet-53, and their shallow features should be clear but their deep features are easy lost after multiple down sampling. So, how to utilize the shallow features as much as possible to improve the detection accuracy becomes the key issue should be solved. This paper proposes a method of feature reusing inspired from DenseNet to achieve this goal.

3.2. Densely Connected Unit. Different with the ResNet, DenseNet solves the vanishing-gradient problem by connecting each layer to every other layer in a feed-forward fashion. As shown in Figure 2, DenseNet is a narrow network, in which each layer accepts inputs from all previous layers and passes the feature maps to all subsequent layers [12]. Each layer reuses the global features and adds only a few new features remaining other features unchanged. This mechanism of feature reuse makes the DenseNet has fewer parameters than traditional convolutional neural networks. In addition, as each layer can access directly the gradient from the loss function, the DenseNet optimizes the information flow and gradient throughout network, which make it easy to be trained and has low overfitting risk on small training data set.

From Figure 2, it can be seen that DenseNet should keep the size of feature maps in all layers consistent to connect each other. However, as we all know, for every deep convolutional neural network, the feature maps must be downsampled multiple times to expand the receptive field gradually and improve computational efficiency. To solve this contradiction, DenseNet divides the network into several densely connected units, and the convolution layers in a densely connected unit keep same size of feature map, while a densely connected unit downsamples to its next unit by average pooling.

As shown in Figure 3, there are two convolution layers in a densely connected unit. The first convolution layer called the bottleneck layer is used to control the number of feature maps. The second convolution layer is used for feature extraction, and the number of convolution kernels in this layer is called growth rate, as it represents the number of new features added into the global features in this layer. Since a densely connected unit only contains new features and reuses the global features input from all previous layers, the number of convolution kernels is much less than that of the ordinary convolution layer. So, the parameters of DenseNet is reduced greatly; in addition, the feature reuse method is also useful to improve its detection accuracy for small object by keeping shallow features in the final global features.

3.3. Backbone Network of the LSDM. For developing a lightweight ship detection model (LSDM), a new backbone network is constructed based on the combination of darknet-53 and DenseNet. Table 1 shows structure of the LSDM backbone network:

In the backbone network, the densely connected unit of DenseNet is used to replace the residual units in DarkNet-53. Within a densely connected unit, the number of convolution kernels in the bottleneck layer is set to 128, and the growth rate of the feature extraction layer is set to 32. That is, for each densely connected unit, the input feature maps will be compressed, firstly, to 128, and then, 32 new feature maps will be added to the global feature. The size of feature maps become smaller as the layer goes deeper, and more feature maps are needed to keep the feature semantic information abundant. That is, as the size of feature maps decreases, more densely connected units are needed to increase their number. Therefore, in the whole backbone network, 5 levels

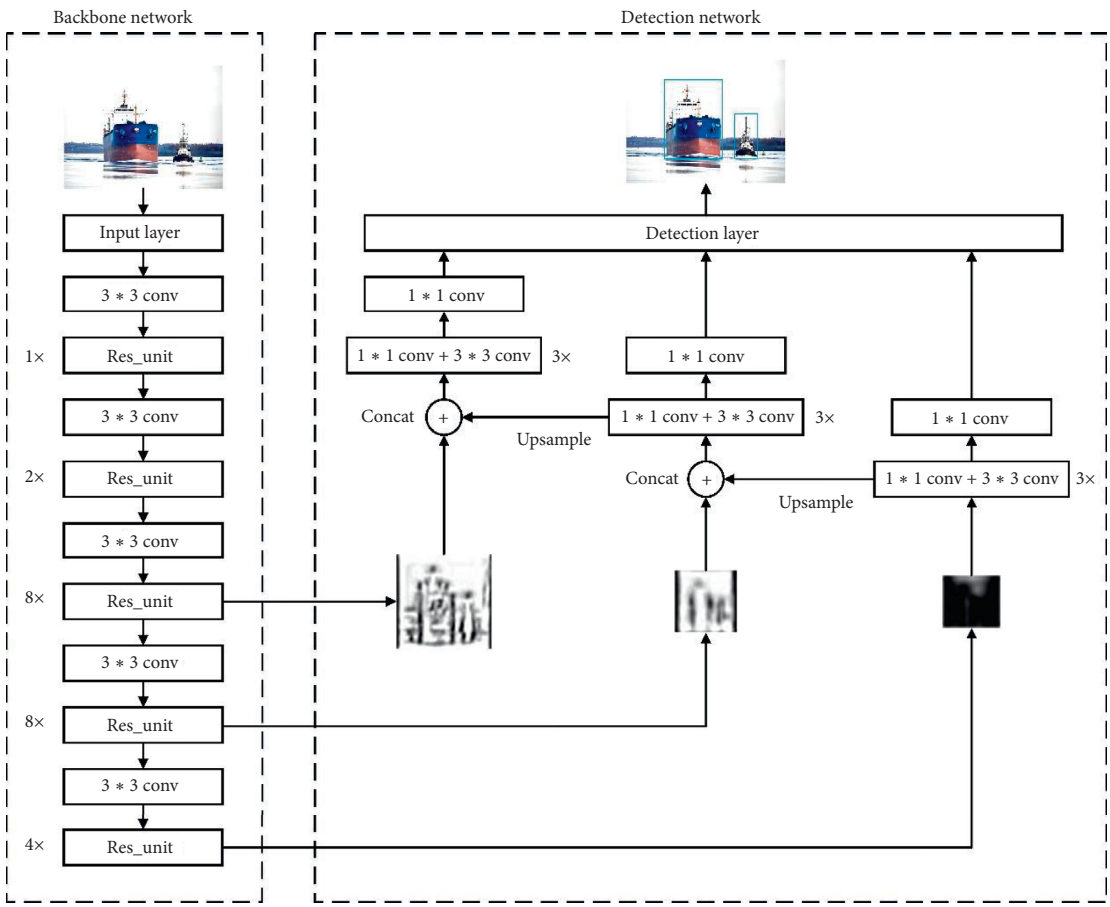


Figure 1: The structure of YOLOv3.

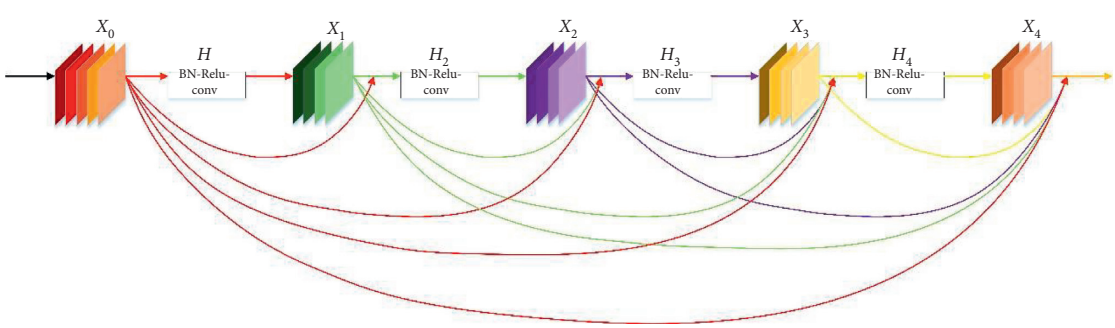


Figure 2: The structure of dense connection.

of densely connected units are adopted with increased numbers of 1, 2, 4, 8, and 16, and the average pooling is used to downsample from one level to its next level. As a result, the backbone network contains 63 convolution layers, and the final number of global feature maps is 1024.

Although there are 11 more layers than Darknet-53, due to the existence of bottleneck layers in densely connected units and feature reuse mechanism, the proposed backbone has fewer parameters than Darknet-53. The parameter number of a convolution layer can be calculated by the following equation:

$$P = k_h \times k_w \times C_{\text{in}} \times C_{\text{out}}, \tag{1}$$

where P is the number of parameters, $k_h \times k_w$ is the size of the kernels, C_{in} is the number of input channels, and C_{out} is the number of output channels.

The parameters of the last residual unit in Darknet-53 and the last densely connected unit in the LSDM backbone network are shown in Table 2. In terms of the whole network, the parameter number of the LSDM backbone network is 3175264, which is just 7.8% of 40549216 that of the Darknet-53.

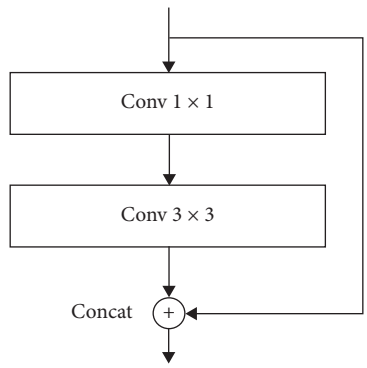


Figure 3: The structure of the densely connected unit.

Table 1: The structure of the LSDM backbone network.

	Type	Filters	Size/stride	Output size
	Conv	32	3*3/1	416*416
	Avg_pool		2/2	208*208
1x	Conv	128	1*1/1	208*208
	Conv	32	3*3/1	208*208
	Avg_pool		2/2	104*104
2x	Conv	128	1*1/1	104*104
	Conv	32	3*3/1	104*104
	Avg_pool		2/2	52*52
4x	Conv	128	1*1/1	52*52
	Conv	32	3*3/1	52*52
	Avg_pool		2/2	26*26
8x	Conv	128	1*1/1	26*26
	Conv	32	3*3/1	26*26
	Avg_pool		2/2	13*13
16x	Conv	128	1*1/1	13*13
	Conv	32	3*3/1	13*13

*The units with gray color are densely connected units (its structure is shown in Figure 3).

Table 2: Parameters comparison between the residual unit and densely connected unit.

Structure	Layer	Kernel size	Input channel	Output channel	Parameters
Residual unit	Conv	1*1	1024	512	524288
	Conv	3*3	512	1024	4718592
Densely connected unit	Conv	1*1	992	128	126976
	Conv	3*3	128	32	36864

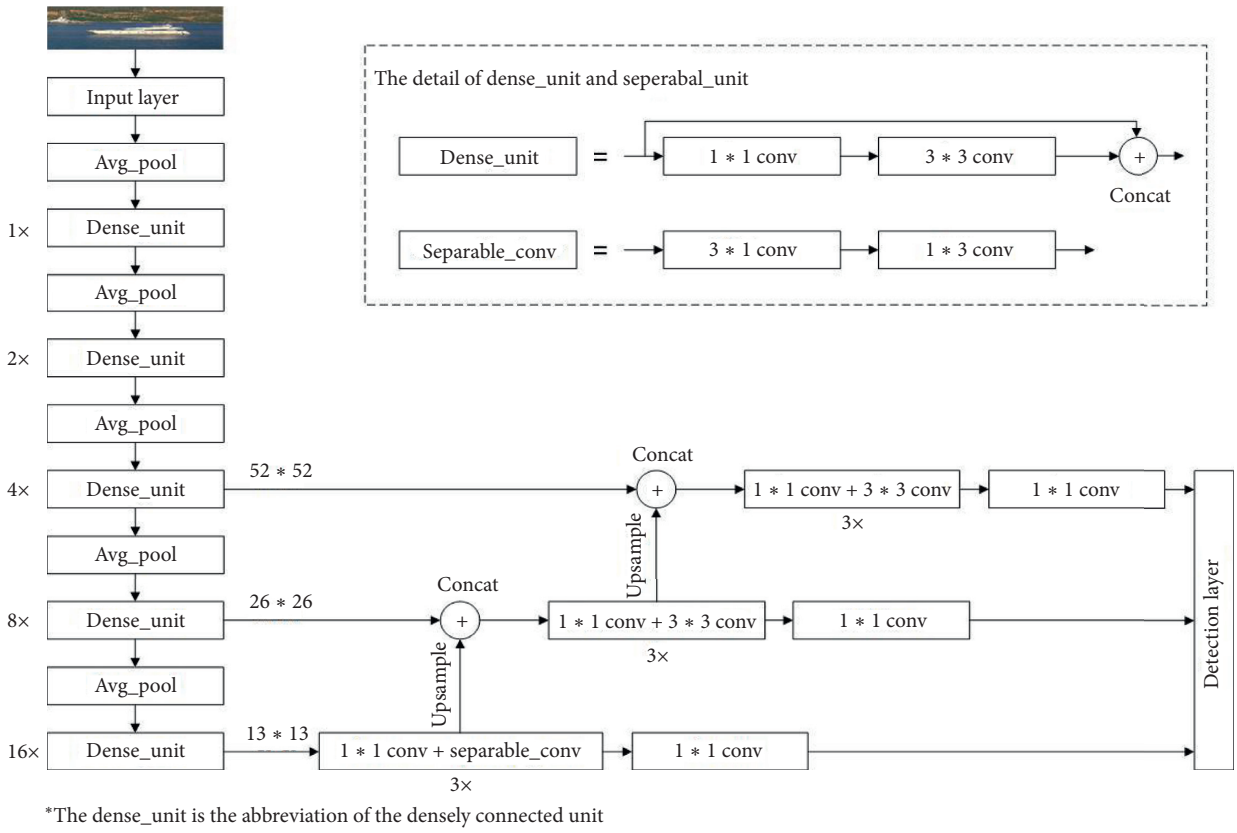
3.4. Backbone Network of LSDM-Tiny. A further compressed backbone network for LSDM-tiny is also investigated. Since the number of global feature maps is related to the number of densely connected units, reducing the number of densely connected units will further decrease the parameters; however, it will also decrease the detection accuracy. In order to keep accuracy as much as possible while reducing densely connected units, a compromise method is applied in the backbone network of LSDM-tiny. There are only two densely connected units no matter which levels; however, a convolution layer with a convolution kernel size of 1*1 is added between level 2 and level 3 and between level 3 and level 4, respectively, to increase the dimension of features. The whole structure of the backbone network of LSDM-tiny is shown in Table 3, and its parameter number is 1291104, which is only 40% of the backbone network of the LSDM and 3% of the Darknet-53.

3.5. LSDM and LSDM-Tiny. The abovementioned backbone networks are, then, used to replace the Darknet-53 on the basis of YOLOv3 to form completed ship detection network LSDM and LSDM-tiny. The overall structure of the LSDM is shown in Figure 4. The output of densely connected units in the last three levels in the backbone network are sent to the right feature pyramid network as inputs. Finally, the three-scale feature maps output by the feature pyramid are used to carry out detection.

Table 3: The structure of the LSDM-tiny backbone network.

	Type	Filters	Size/stride	Output size
1x	Conv	32	3*3/1	416*416
	Conv	128	1*1/1	416*416
	Conv	32	3*3/1	416*416
	Avg_pool		2/2	208*208
2x	Conv	128	1*1/1	208*208
	Conv	32	3*3/1	208*208
	Conv	256	1*1/1	208*208
	Avg_pool		2/2	104*104
2x	Conv	128	1*1/1	104*104
	Conv	32	3*3/1	104*104
	Conv	640	1*1/1	104*104
	Avg_pool		2/2	52*52
2x	Conv	128	1*1/1	52*52
	Conv	32	3*3/1	52*52
	Avg_pool		2/2	26*26
2x	Conv	128	1*1/1	26*26
	Conv	32	3*3/1	26*26
	Avg_pool		2/2	13*13
2x	Conv	128	1*1/1	13*13
	Conv	32	3*3/1	13*13

*The units with gray color are densely connected units (their structure is shown in Figure 3).



*The dense_unit is the abbreviation of the densely connected unit

Figure 4: The overall structure of the LSDM network.

In addition to the backbone network, the feature pyramid network is also improved. The 3*3 convolution layer of the last branch is split into a combination of 3*1 and 1*3 convolution layers which can be called spatial separation convolution. Spatial separation convolution can reduce the parameter number by half, and it will perform best in the last branch because there are mostly small-scale feature maps. Table 4 shows the parameters comparison between the

Table 4: Parameters comparison of standard convolution and spatial separation convolution.

Structure	Layer	Kernel size	Input channel	Output channel	Parameters
Conv2d	Conv	3*3	512	1024	4718592
Separable conv	Conv	3*1	512	512	786432
	Conv	1*3	512	1024	1572864

Figure 5: Samples of the training dataset.

Table 5: Comparison of model training results.

	Recall	Precision	AP	$F1$	Parameters (M)	FPS
LSDM	0.954	**0.851**	**0.940**	**0.899**	20.02	16.97
YOLOv3	0.946	0.830	0.930	0.884	61.47	**13.48**
LSDM-tiny	**0.958**	0.804	0.935	0.874	7.05	27.09
YOLOv3-tiny	**0.916**	**0.707**	**0.881**	**0.798**	8.66	**29.49**

standard convolution and spatial separation convolution in the feature pyramid network.

In summary, by using new backbone network and spatial separation convolution, the total parameters of the LSDM is 20022112, which is about 32% of YOLOv3.

LSDM-tiny can be transformed from the LSDM by replacing the backbone network with that for LSDM-tiny (As shown in Table 3) and deleting the branch on the 52*52 scale in the feature pyramid network. The total parameters of LSDM-tiny are 7054368, which is about 36% of the LSDM and about 12% of YOLOv3.

3.6. Model Training Details. In order to improve the detection accuracy of the LSDM and LSDM-tiny and speed up their training speed, some tricks also can be added. Firstly, the LeakyReLU activation function is used to replace ReLU (Rectified Linear Units), and the negative half-axis slope of the function is set to 0.1; its formula is shown in the following equation:

$$f(x_i) = \begin{cases} x_i, & x_i \geq 0, \\ \dfrac{x_i}{a}, & x_i < 0, a = 0.1. \end{cases} \tag{2}$$

Secondly, to make the models converge faster, momentum is added to the SGD optimizer, and the improved optimization function is shown in the following equation:

$$\begin{aligned} v &= \beta \times v - \alpha \times dx, \\ x &= x + v, \end{aligned} \tag{3}$$

where β is the momentum parameter set to 0.949, and the initial value of variable v is set to 0.

Thirdly, in order to reduce the risk of overfitting, weight decay for the parameters of the convolution layers is used, and its attenuation coefficient is set to 0.000489.

4. Experiments and Analysis

4.1. Ship Image Dataset. The ship image dataset contains 2270 pictures captured by web crawlers and processed by data enhancement methods such as random image flipping, noise addition, and color enhancement. Also, its annotation file contains ship category information (only ships here, labeled 0) and normalized bounding box coordinate information. 80% of the dataset is used to train models, and 20% is used for testing. Figure 5 shows some samples of the training dataset.

4.2. Experiment. The LSDM and LSDM-tiny are implemented by PyTorch, and their performances are investigated and compared with YOLOv3 and YOLOv3-tiny on the abovementioned dataset under an NVIDIA GTX1060 (3 g) environment. The evaluation indicators include precision, AP (Average Precision), recall, $F1$, parameter number, and FPS (Frames Per Second). The value of FPS is the average

(a) (b)

Figure 6: Detection effect comparison: (a) single ship detection; (b) multiple ships detection.

number of detected images per second obtained by detecting 1000 images through the detection network.

The experiment results are shown in Table 5:

Figure 6 shows the detection effect of the four models, LSDM-tiny, YOLOv3-tiny, LSDM, and YOLOv3. Figure 6(a) is the case for single ship detection, and Figure 6(b) is the case for multiple ships detection.

4.3. Analysis. The abovementioned results show that the LSDM has higher FPS than YOLOv3. As expected, with only one-third parameters, the LSDM is faster than YOLOv3 when detecting ships. More importantly, the LSDM also has better performances in "Recall," "Precision," "AP," and "F1" than YOLOv3. That is, the LSDM has higher accuracy than YOLOv3.

It can be clearly observed in the detection effect image for single ship detection in the Figure 6(a) that the LSDM is superior to other models in terms of positioning accuracy and classification effect due to its deeper network depth. For multiple ship detection in the Figure 6(b), it also can be observed that the LSDM's positioning is the most accurate and has no problems of missed detection and misdetection happened in other models due to its retaining of shallow features. From the comparison results, the effect of feature reusing in densely connected units is validated.

The abovementioned results also show that LSDM-tiny is much faster than the LSDM and YOLOv3 as expected. The "FPS" of LSDM-tiny is about double that of YOLOv3, but a bit less than that of YOLOv3-tiny. However, LSDM-tiny has a higher accuracy than YOLOv3-tiny. It can also be clearly observed in the detection effect image in Figure 6 that the densely connected units of LSDM-tiny make the classification score and positioning accuracy better than YOLOv3-tiny both in single ship and multiple ship detection cases. This comparison results further validate that it is important for improving detection accuracy of a small object to keeping shallow features into the final global features.

5. Conclusions

This paper proposes a lightweight ship detection model (LSDM) based on YOLOv3 and DenseNet. In the LSDM, the features of the shallow layer are allow to be retained and used in subsequent layers. This mechanism reduces, greatly, the parameters and optimizes the structure of the backbone network, and spatially separated convolution is used to further reduce the parameters in the feature pyramid network. The two improvements make the parameters of the LSDM be only one-third of YOLOv3. Also, the experimental results show that the LSDM is not only faster than YOLOv3 but also has higher accuracy.

Furthermore, a model called LSDM-tiny is constructed as a simple version of the LSDM. By reducing the number of densely connected units, the parameters of LSDM-tiny is only one-eighth of YOLOv3. The experimental results show that the detection speed of LSDM-tiny is about double that of YOLOv3, with losing a little accuracy. Also, comparing with YOLOv3-tiny, LSDM-tiny has a similar detection speed but has a higher accuracy due to the reuse mechanism of feature maps.

The LSDM and LSDM-tiny are proposed for fast ship detection on existing normal even poor hardware condition. In the future, two aspects will be studied further. First, for the problem of uneven detection of positive and negative samples in YOLOv3, how to add a stricter penalty mechanism to reduce the impact of negative samples will be studied. Secondly, to detect a small ship object in the camera images, how to increase multiscale detection channels while maintaining a small number of parameters will be studied.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by "the Fundamental Research Funds for the Central Universities" under Grant 3132019400.

References

[1] X. Chen, L. Qi, Y. Yang et al., "Video-based detection infrastructure enhancement for automated ship recognition and behavior analysis," *Journal of Advanced Transportation*, vol. 2020, Article ID 7194342, 12 pages, 2020.

[2] X. Chen, H. Chen, H. Wu et al., "Robust visual ship tracking with an ensemble framework via multi-view learning and wavelet filter," *Sensors*, vol. 20, no. 3, p. 932, 2020.

[3] I. Im, D. Shin, and J. Jeong, "Components for smart autonomous ship architecture based on intelligent information technology," *Procedia Computer Science*, vol. 134, pp. 91–98, 2018.

[4] J.-Y. Zhuang, L. Zhang, S.-Q. Zhao, J. Cao, B. Wang, and H.-B. Sun, "Radar-based collision avoidance for unmanned surface vehicles," *China Ocean Engineering*, vol. 30, no. 6, pp. 867–883, 2016.

[5] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, December 2015.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[7] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Venice, Italy, October 2017.

[8] W. Liu, D. Anguelov, D. Erhan et al., "Single shot MultiBox detector," in *Proceedings of the Computer Vision—ECCV 2016*, pp. 21–37, Amsterdam, The Netherlands, October 2016.

[9] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR), pp. 779–788, Las Vegas, NV, USA, June 2015.

[10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[11] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "AlexNet-level accuracy with 50x fewer parameters and <1MB model size," ArXiv 2017, abs/1602.07360, 2017.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, Honolulu, HI, USA, July 2017.

[13] A. G. Howard, M. Zhu, B. Chen et al., "Mobile nets: efficient convolutional neural networks for mobile vision applications," ArXiv 2017, abs/1704.04861, 2017.

[14] X. Zhang, X. Zhou, M. X. Lin, and J. Sun, "ShuffleNet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018)*, New York, NY, USA, June 2018.

[15] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," ArXiv 2018, abs/1804.02767, 2018.

[16] W. Fang, F. Zhang, V. Sheng, and Y. Ding, "A method for improving CNN-based image recognition using DCGAN," *Computers, Materials & Continua*, vol. 57, no. 1, pp. 167–178, 2018.

[17] R. H. Meng, S. G. Rice, J. Wang, and X. Sun, "A fusion steganographic algorithm based on faster R-CNN," *CMC-Computers Materials & Continua*, vol. 55, no. 1, pp. 1–16, 2018.

[18] Q. Cui, S. Mclntosh, and H. Y. Sun, "Identifying materials of photographic images and photorealistic computer generated graphics based on deep CNNs," *CMC-Computers Materials & Continua*, vol. 55, pp. 229–241, 2018.

[19] Z. Dong and B. Lin, "Learning a robust CNN-based rotation insensitive model for ship detection in VHR remote sensing images," *International Journal of Remote Sensing*, vol. 41, no. 9, pp. 3614–3626, 2020.

[20] X. Yang, H. Sun, X. Sun, M. Yan, Z. Guo, and K. Fu, "Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network," *IEEE Access*, vol. 6, pp. 50839–50849, 2018.

[21] W. Fan, F. Zhou, X. Bai, M. Tao, and T. Tian, "Ship detection using deep convolutional neural networks for PolSAR images," *Remote Sensing*, vol. 11, no. 23, p. 2862, 2019.

[22] S. Zhang, R. Wu, K. Xu, J. Wang, and W. Sun, "R-CNN-Based ship detection from high resolution remote sensing imagery," *Remote Sensing*, vol. 11, no. 6, p. 631, 2019.

[23] J. Jiao, Y. Zhang, H. Sun et al., "A densely connected end-to-end neural network for multiscale and multiscene SAR ship detection," *IEEE Access*, vol. 6, pp. 20881–20892, 2018.

[24] K. Kim, S. Hong, B. Choi, and E. Kim, "Probabilistic ship detection and classification using deep learning," *Applied Sciences*, vol. 8, no. 6, p. 936, 2018.

[25] Q. An, Z. Pan, L. Liu, and H. You, "DRBox-v2: an improved detector with rotatable boxes for target detection in SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 8333–8349, 2019.

[26] B. Liu, S. Wang, J. Zhao, and M. Li, "Ship tracking and recognition based on Darknet network and YOLOv3 algorithm," *Journal of Computer Applications*, vol. 39, pp. 1663–1668, 2019.

[27] Y. Zhang, J. Shu, L. Hu, Q. Zhou, and Z. Du, "A ship target tracking algorithm based on deep learning and multiple features," *Society of Photographic Instrumentation Engineers*, vol. 11433, 2020.

[28] Y. Wang, C. Wang, and H. Zhang, "Combining a single shot multibox detector with transfer learning for ship detection using sentinel-1 SAR images," *Remote Sensing Letters*, vol. 9, no. 8, pp. 780–788, 2018.

[29] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "Automatic ship detection based on RetinaNet using multi-resolution gaofen-3 imagery," *Remote Sensing*, vol. 11, no. 5, p. 531, 2019.

[30] Y.-L. Chang, A. Anagaw, L. Chang, Y. Wang, C.-Y. Hsiao, and W.-H. Lee, "Ship detection based on YOLOv2 for SAR imagery," *Remote Sensing*, vol. 11, no. 7, p. 786, 2019.

[31] T. Zhang and X. Zhang, "High-speed ship detection in SAR images based on a grid convolutional neural network," *Remote Sensing*, vol. 11, no. 10, p. 1206, 2019.

[32] L. Qi, B. Y. Li, L. K. Chen et al., "Ship target detection algorithm based on improved faster R-CNN," *Electronics*, vol. 8, no. 9, p. 959, 2019.

[33] Z. Lin, K. Ji, X. Leng, and G. Kuang, "Squeeze and excitation rank faster R-CNN for ship detection in SAR images," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 5, pp. 751–755, 2019.

[34] Z. X. Zhang, H. L. Li, G. Q. Zhang et al., "A high-speed cascaded convolutional neural network for ship detection with multispectral images," *Journal of Infrared and Millimeter Waves*, vol. 38, no. 3, pp. 290–295, 2019.

[35] T. W. Zhang, X. L. Zhang, J. Shi, and S. J. Wei, "Depthwise separable convolution neural network for high-speed SAR ship detection," *Remote Sensing*, vol. 11, no. 21, p. 2483, 2019.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 3490536, 14 pages
https://doi.org/10.1155/2020/3490536

Research Article

Research on Credit Risk Measurement of Small and Micro Enterprises Based on the Integrated Algorithm of Improved GSO and ELM

Li Jingming,[1] Li Xuhui,[1] Dai Daoming,[1] Ruan Sumei,[2] and Zhu Xuhui[3]

[1]*School of Management Science and Engineering, Anhui University of Finance and Economics, Bengbu 233030, Anhui, China*
[2]*School of Finance, Anhui University of Finance and Economics, Bengbu 233030, Anhui, China*
[3]*School of Management, Hefei University of Technology, Hefei 230009, China*

Correspondence should be addressed to Li Jingming; homer998@163.com and Li Xuhui; lixh122@163.com

Received 24 April 2020; Revised 26 August 2020; Accepted 4 September 2020; Published 18 September 2020

Academic Editor: Chunjia Han

Small and micro enterprises play a very important role in economic growth, technological innovation, employment and social stability etc. Due to the lack of credible financial statements and reliable business records of small and micro enterprises, they are facing financing difficulties, which has become an important factor hindering the development of small and micro enterprises. Therefore, a credit risk measurement model based on the integrated algorithm of improved GSO (Glowworm Swarm Optimization) and ELM (Extreme Learning Machine) is proposed in this paper. First of all, according to the growth and development characteristics of small and micro enterprises in the big data environment, the formation mechanism of credit risk of small and micro enterprises is analyzed from the perspective of granularity scaling, cross-border association and global view driven by big data, and the index system of credit comprehensive measurement is established by summarizing and analyzing the factors that affect the credit evaluation index. Secondly, a new algorithm based on the parallel integration of the good point set adaptive glowworm swarm optimization algorithm and the Extreme learning machine is built. Finally, the integrated algorithm based on improved GSO and ELM is applied to the credit risk measurement modeling of small and micro enterprises, and some sample data of small and micro enterprises in China are collected, and simulation experiments are carried out with the help of MATLAB software tools. The experimental results show that the model is effective, feasible, and accurate. The research results of this paper provide a reference for solving the credit risk measurement problem of small and micro enterprises and also lay a solid foundation for the theoretical research of credit risk management.

1. Introduction

In recent years, China's economy has maintained a good momentum of development. The number of domestic enterprises has grown steadily, especially small and micro enterprises, which have become a large number of dynamic enterprise groups in the main body of market economy. So small and micro enterprises are an important part of China's economy. However, small and micro enterprises also face severe financing difficulties in the process of survival and development. Because of their own reasons and political, economic, legal, and other external factors, they are in a dilemma of few financing channels and high financing costs. From a macro perspective, China's market economy system is still not perfect, the credit system is defective, small and micro enterprise groups cannot get a comprehensive and objective credit evaluation, and the financial investment industry has not paid enough attention to it, resulting in narrow financing channels. From the micro point of view, small and micro enterprises have light assets, small scale, and unknown financial situation, so it is difficult to reasonably assess the credit risk. Also, the private lending interest rate and cost are high, which makes financing a difficult problem.

The measurement of credit risk of small and micro enterprises is a research hotspot of scholars at home and abroad. In China, Man et al. [1] constructs lasso logistic

model to identify the key indicators that affect the credit risk of small and micro enterprises based on the data of 496 unlisted small and micro enterprises' default in the process of bank loan. Han [2] uses the fuzzy integral support vector machine algorithm to build the credit risk assessment model of small and micro enterprises from the perspectives of loan mode, financial situation, and characteristics of enterprises. Su [3] uses the mixed analysis method of soft information and hard information to measure the credit risk of small and micro enterprises. Zhang et al. [4] proposes a dynamic combination evaluation model based on fuzzy clustering and SOM-k algorithm to evaluate the credit risk of small and micro enterprises. Based on the theory of information asymmetry, Yoshino and Taghizadeh-hesary [5] built a theoretical model of SMEs' credit and empirically analyzed the influencing factors of the optimal guarantee ratio of enterprises. Hanggraeni et al. [6] proposes a more suiTABLE credit risk measurement method for small- and medium-sized enterprises, which uses the method of default probability measure to improve Z value and estimate its critical value. Based on the fuzzy matter-element Euclidean approach degree model, Zhou et al. [7] elaborated the inherent law of the credit risk evaluation of the industrial park enterprises and provided a practical method for the research of the industrial park enterprise credit rating evaluation. Shi et al. [8] discovered that the default of small and medium-sized enterprises with high-grade loans will lead to low recovery rate, so they used three credit data sets to test a credit risk method to help small- and medium-sized enterprises solve the mismatch problem.

In conclusion, the current research states of credit risk measurement methods of small and micro enterprises are discussed from different perspectives of constructing credit evaluation index system and risk measurement model. Among them, foreign scholars emphasize on quantitative research by mining the credit data of some small- and medium-sized enterprises for research and evaluation. Domestic scholars pay more attention to qualitative research on the construction of indicator system and quantitative research on credit risk measurement. In addition, big data provides more comprehensive, accurate, and precise digital management for many subject areas. Chen and Wu [9] believes that traditional management has become data management and traditional decision-making has become decision-making based on data mining analysis in the background of big data. The research of big data application should start from the data characteristics, problem characteristics, and management decision characteristics of big data. A credit risk measurement model for small and micro enterprises driven by big data is proposed in this background.

First of all, the characteristics of small and micro enterprises and the impact of big data on small and micro enterprises' credit evaluation are analyzed in depth from the perspective of granularity scaling, cross-border association, and global view driven by big data, and the mechanism of small and micro enterprises' credit risk formation is explored. On this basis, credit risk measurement indicators are selected to establish a credit risk measurement indicator system. Secondly, due to the non-linear inherent relationship of credit risk data, adaptive learning characteristics of neural network have become the most common and relatively accurate classifier in credit risk measurement [10–13].

Therefore, ELM feedforward neural network is employed to solve the problem of credit risk measurement. Because the number of hidden layer nodes has a great influence on the classification accuracy, the initial weight, threshold, and hidden layer node parameters of ELM are optimized by the improved GSO algorithm. A parallel integrated learning algorithm based on the improved GSO algorithm and ELM neural network is applied to the credit risk measurement of small and micro enterprises for solving the problem of credit risk measurement of small and micro enterprises.

The research in this paper mainly includes 3 parts. Section 2 gives explanations on relevant algorithms and theories, which puts forward the improved GSO algorithm based on good point set and combines the improved GSO algorithm with adaptive step-size strategy. Section 3 introduces the index system of credit comprehensive measurement established by summering and analyzing the factors from the perspective of granularity scaling, cross-border association, and global view driven by big data and proposes a credit risk measurement model based on the integrated algorithm of IGSO and ELM. Section 4 selects 10 Banchmarks standard functions to evaluate the performance of IGSO algorithm. Collect data set of micro small enterprise from a region in China to verify credit risk measurement model of micro small enterprise based on the integrated algorithm of IGSO and ELM.

2. Description of Relevant Algorithms

2.1. GSO Algorithm. Metaheuristic algorithm is an improvement of heuristic algorithm, which is the combination of random algorithm and local search algorithm. Traditional metaheuristic algorithms include tabu search algorithm, simulated annealing algorithm, genetic algorithm, ant colony optimization algorithm, particle swarm optimization algorithm, artificial fish swarm algorithm, artificial bee colony algorithm, artificial neural network algorithm, glowworm swarm optimization algorithm, etc. At present, there are several new metaheuristic algorithms, such as Henry gas solubility (HGS) [14], Slime mould algorithm (SMA) [15], and Harris hawk optimization (HHO) [16]. Swarm intelligence optimization algorithm is an algorithm designed based on the group behavior characteristics of natural organisms, and it is also a method to solve distributed problems. The common swarm intelligence algorithms are ant colony algorithm, particle swarm optimization algorithm, artificial fish swarm algorithm, artificial bee colony algorithm, wolf colony algorithm, glowworm swarm optimization algorithm, and so on. GSO (Glowworm swarm optimization) algorithm is not only a metaheuristic algorithm, but also a swarm intelligence algorithm.

GSO algorithm [17] is also a bionic swarm intelligence algorithm proposed by Indian scholars Krishnand and Ghose in 2005, which simulates the foraging and courtship

behavior of swarms in nature. Comparing with other intelligent optimization algorithms, GSO algorithm is simple and easy to implement with clear algorithm flow. Due to the advantages of less parameter setting, simple working mechanism, easy programming, and multi extreme value capture, this paper selects the GSO algorithm as the research object to carry out the relevant theoretical and application research. The algorithm has been currently applied to multimode function optimization [18], multisource signal location [19], wireless sensor node layout [20], simulation robot group [21], cluster analysis [22] and other fields, which show a good research and application prospects.

The main idea of GSO algorithm is that the glowworm in search space represents every feasible solution of the optimization problem. Each glowworm has its own fluorescein and sensing radius. Its brightness is related to the target value of its location. The glowworm with higher brightness has better index value. In the iterative process, the brighter glowworm has stronger attraction ability, which attracts other glowworms to move towards it. Since each glowworm has its own decision radius, the decision radius will be affected by neighboring glowworms at the same time. When the number of glowworms around it is few, the decision radius of glowworms will increase, which can attract more glowworms around. When there are more glowworms around, the decision radius will be smaller. At last, most of the glowworms will gather in several positions with better objective function value to reach the best value. GSO algorithm is described by the following mathematical formulas:

$$l_i(t) = (1-\rho)l_i(t-1) + \gamma J(x_i(t)), \tag{1}$$

$$N_i(t) = \left\{ j\colon \left\| x_j(t) - x_i(t) \right\| < r_d^i; l_i(t) < l_j(t) \right\}, \tag{2}$$

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k\in N_i(t)} l_k(t) - l_i(t)}, \tag{3}$$

$$x_i(t+1) = x_i(t) + s * \left(\frac{x_j(t) - x_i(t)}{\left\| x_j(t) - x_i(t) \right\|} \right), \tag{4}$$

$$r_d^i(t+1) = \min\left\{ r_s, \max\left\{ 0, r_d^i(t) + \beta\left(n_t - \left|N_i(t)\right|\right) \right\} \right\}, \tag{5}$$

where $l_i(t)$ is the fluorescein value of glowworm in the t-th iteration, $J(x_i(t))$ is the target function value in the t-th iteration, ρ is the volatilization coefficient of fluorescein value, γ is the enhancement factor of fluorescein, β is the coefficient of change of perception radius, n_t is the number threshold of neighborhood glowworms, r_s is perception radius of neighborhood, r_d is the dynamic decision domain, and s is the step size.

The GSO algorithm is described as follows:

Step 1: Initializing the relevant parameters of the algorithm, including population size, iteration times, and other parameters, to be set.

Step 2: The objective function value $J(x_i(t))$ corresponding to the position $x_i(t)$ of glowworm i in the t-th iteration is converted to the fluorescein value $l_i(t)$ by formula (1).

Step 3: Within the radius of its dynamic decision domain $r_d^i(t)$ $(0 < r_d^i < r_s)$, each glowworm chooses individuals whose brightness is higher than itself to form its domain set $N_i(t)$ by formula (2). The probability $p_{ij}(t)$ of individuals i moving to the individuals $j(j \in N_i(t))$ in their domain set is calculated by formula (3).

Step 4: Selecting the object to move and updating the position $x_i(t)$ of glowworm i according to formula (4).

Step 5: Updating the dynamic decision domain radius $r_d^i(t)$ of glowworm i according to formula (5).

Step 6: Judging whether the algorithm reaches the maximum iterations or not. If not, turn to step 2; otherwise, end.

Due to the uneven distribution of the initial solution in the solution space, the algorithm is unstable, slow in convergence speed, and low in accuracy. In order to avoid the premature problem of GSO algorithm, the idea of Good Point Set (GPS) Theory is employed to generate the initial glowworm population with uniform distribution. At the same time, a new inertia weight function of glowworm moving is used to dynamically update the moving step length, i.e., adaptive step-size, so as to further improve the stability, convergence speed, and accuracy of the GSO algorithm.

2.2. Improved GSO Algorithm

2.2.1. Good Point Set Improved GSO Algorithm. The theory of good point set [23] was put forward by Luogeng and Wang in the book of number theory in approximate analysis. Good point set can provide better support for the calculation in high-dimensional spaces [24]. In [25], a glowworm optimization algorithm based on the good point set is proposed to optimize the initial weight and threshold of BP neural network, which is employed to solve the problem of agricultural drought evaluation. According to [23–25], the basic definition and structure of good point set are described as follows.

Definition 1. Let G_m be a unit cube in m-dimensional Euclidean space, which is expressed as $\langle x \rangle = (x_1, x_2, \ldots, x_m)$, $i = 1, 2, \ldots, m$, $0 \le x_i \le 1$.

Definition 2. According to number and dimension of the sample G_m, a point set of the same size is generated as the target set $P_n(i) = \{(x_1^n(i), x_2^n(i), \ldots, x_m^n(i)), 1 \le i \le n\}$, where $x_j^n(i) = c(e^j * i), 1 < j < m$, and $c(k)$ is the decimal part of k. Compare the sample points $\langle r \rangle = (r_1, r_2, \ldots, r_m)$ with the target set $p_n(i)$, and Assume $\varphi(n) = \sup|N_n(r)/n - |r||$, where $r \in G_m$, $N_n(\langle r \rangle) = N_n(r_1, r_2, \ldots, r_m)$, $|\langle r \rangle| = r_1, r_2, \ldots, r_m$, and $\varphi(n)$ is known as the deviation of point set $p_n(k)$.

Definition 3. Set $\varphi(n) = C(\langle r\rangle, \varepsilon)n^{(-1+\varepsilon)}$, where $C(\langle r\rangle, \varepsilon)n^{(-1+\varepsilon)}$ is a constant related to $\varepsilon(\varepsilon > 0)$ and r. $\varphi(n)$ is also regarded as the deviation of $P_n(k)$, and $P_n(k) = \{(x_1^n * k, x_2^n * k, \ldots, x_m^n * k), k = 1, 2, \ldots, n\}$. So the $p_n(k)$ is considered as a good point set and r is a good point.

It can be seen from [25] that the order of error is only related to the number of samples and independent of the spatial dimension of samples, which provides a very good algorithm for high-dimensional approximation. The method based on good point set theory is better than the random method to generate the initial population, because the deviation of the good point set method is $O(n^{-1+\varepsilon})$, and the deviation of the random method is $\varphi(n) = O(n^{-1/2} \log\log n)^{1/2}$.

Because of using the theory of good point set to construct the initial population of glowworm, its calculation accuracy is independent of the dimension. So using the method based on good point set theory to design the initial glowworm population uniformly can overcome the shortcomings of the traditional methods and can produce the initial population with better diversity.

The number of initial glowworm population is n, and the method of taking the set of n good points in the m-dimensional space is defined as follows:

(1) Generating good point set by exponential sequence method: $r_k = \{e^k, 1 \le k \le s\}$.

(2) Generating good point set by square root sequence method: $r_k = \sqrt{p_k}$, $1 \le k \le m$.

(3) Generating good point set by circle Division method: $r_k = (2\cos 2\pi k/p)$, $1 \le k \le m$.

Using the good point set method to design the initial glowworm population uniformly can produce the initial population with better diversity. Figures 1–4 show the two-dimensional initial population distribution charts with the scale of 400 generated by the random method and the three good point set methods respectively. It can be seen directly from the figures that the data point distribution in good point set method is much more uniform than that of random points. The construction of the good points is independent of the space dimension and can be used to solve the high-dimensional problems. Because the number of points taken by the good point set method is the same, the distribution effect is the same. Therefore, a relatively good initial glowworm population can be obtained by mapping the generated good points to the target solution space of GSO algorithm.

2.2.2. Adaptive Step-Size Improved GSO Algorithm. In the GSO algorithm, each glowworm has a different search range determined by the sensing radius. GSO algorithm can find the global or local optimal solution, which depends on whether the individual can move within the sensing range. With the increase of the number of iterations, glowworm individuals tend to converge near the peak. At this time, if the distance between glowworm individuals and the peak is less than the moving step, the individual will move to the other side of the peak. If the iteration is repeated again, the glowworm individual will move to the other side of the peak. The individual still fails to reach the optimal peak at this time. The glowworm individual repeatedly moves around the peak, which is called oscillation phenomena. To solve this

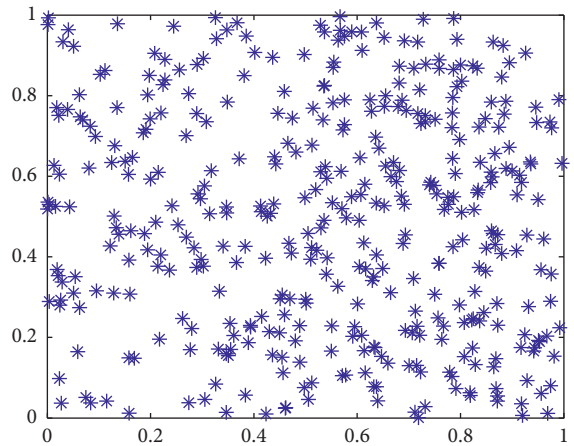


Figure 1: Initial population generated by the random method.

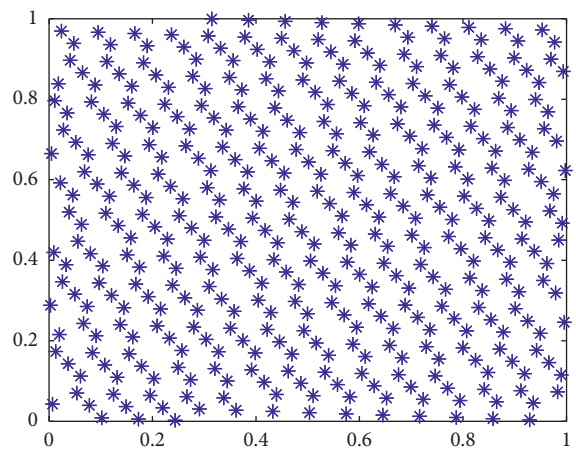


Figure 2: Initial population generated by exponential sequence method.

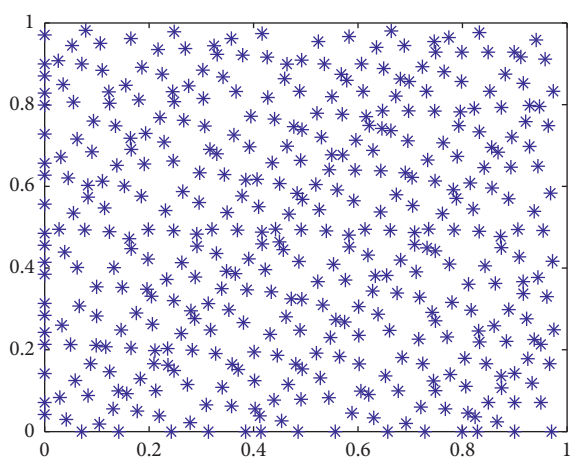


Figure 3: Initial population generated by square root sequence method.

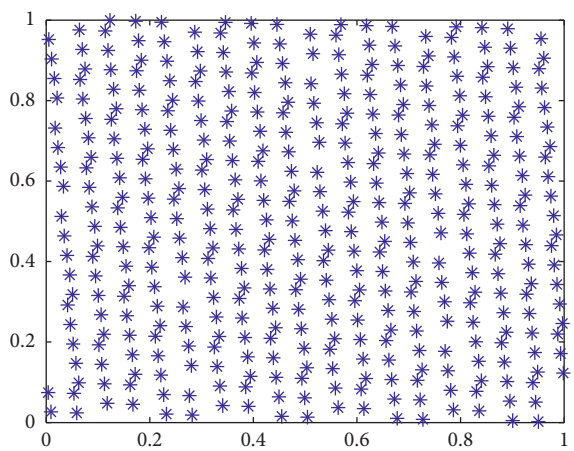


Figure 4: Initial population generated by circle Division method.

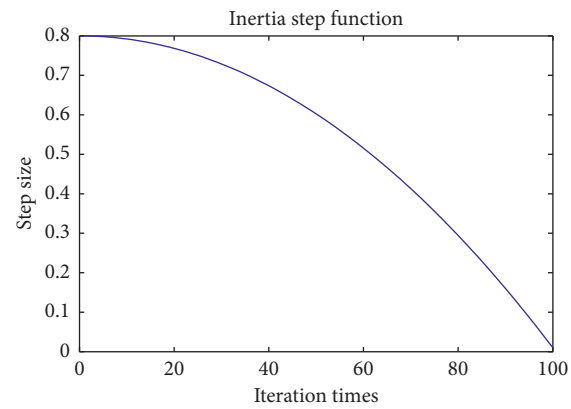


Figure 5: Inertia weight function of glowworm moving step.

problem, it is necessary to adjust the step size dynamically according to the search results of different stages, so as to deal with the relationship among the global optimization ability, convergence speed, and optimization accuracy. Based on the idea of inertial weight of particle velocity in particle swarm optimization algorithm [26], the inertial weight function of glowworm displacement $s(t)$ is defined, which is the function of iterations t. The inertia weight function of moving step size is specifically defined as

$$s(t) = s_1 - (s_1 - s_2) * \left(\frac{t}{T_{\max}}\right)^2, \tag{6}$$

where s_1 is the minimum moving step size, s_2 is the maximum moving step size, and $T_{\max}$ is the maximum number of iterations.

The inertia weight function of glowworm moving step is shown in Figure 5. Therefore, the position update of the i-th glowworm is no longer carried out by formula (4) but by formula (7) in improved GSO algorithm:

$$x_i(t+1) = x_i(t) + s(t) * \left(\frac{x_j(t) - x_i(t)}{\left\|x_j(t) - x_i(t)\right\|}\right). \tag{7}$$

2.3. ELM. The traditional single hidden layer feedforward neural network model is mainly based on the algorithm of gradient descent, such as BP neural network algorithm. Its learning speed is difficult to meet the needs, and it is easy to lead to local optimal solution. In different application scenarios, parameters need to be adjusted.

In 2004, Huang et al. [27] proposed a new feedforward neural network, which is the extreme learning method (ELM). The ELM has some advantages in solving data fitting, regression, classification, pattern recognition, and other related problems [28] and has been widely used in cx-image processing [29], medical diagnosis [30], fault inspection [31] traffic sign recognition [32], and other fields. The ELM is a fast learning single-layer feedforward neural network algorithm based on Moore Penrose matrix theory.

Suppose the number of neurons in the hidden layer of ELM is L and the number of training samples is N. There are N arbitrary samples (X_i, t_i), where $X_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \in R^n$, $t_i = [t_{i1}, t_{i2}, \ldots, t_{im}] \in R^m$. A single hidden layer neural network with L hidden layer nodes can be expressed as

$$o_j = \sum_{i=1}^{L} \beta_i g\left(W_i \cdot X_j + b_i\right), \quad j = 1, 2, \ldots, N, \tag{8}$$

where $g(x)$ is the activation function, $W_i = (w_{i1}, w_{i2}, \ldots, w_{in})$ represents the input weight, $\beta_i = (\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}) \in R^m$ is the output weight, b_i is the offset of the i-th hidden layer unit, and $W_i \cdot X_j$ is Inner product of W_i and X_j.

The learning goal of single hidden layer neural network is to minimize the output error, which can be expressed as

$$\sum_{j=1}^{L} \left\|o_j - t_i\right\| = 0. \tag{9}$$

That is, there are β_i, W_i, and b_i, making

$$\sum_{i=1}^{L} \beta_i g\left(W_i \cdot X_j + b_i\right) = t_j, \quad j = 1, 2, \ldots, N. \tag{10}$$

It can also be expressed as

$$H\beta = T, \tag{11}$$

where H is the output of the hidden layer node; β is the output weight; and t is the expected output.

$$H = \begin{bmatrix} g(W_1 \cdot X_1 + b_1) & g(W_2 \cdot X_1 + b_2) & \cdots & g(W_L \cdot X_1 + b_L) \\ g(W_1 \cdot X_2 + b_1) & g(W_2 \cdot X_2 + b_2) & \cdots & g(W_L \cdot X_2 + b_L) \\ \vdots & \vdots & \vdots & \vdots \\ g(W_1 \cdot X_N + b_1) & g(W_2 \cdot X_N + b_2) & \cdots & g(W_L \cdot X_N + b_L) \end{bmatrix}_{N \times L}, \tag{12}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \quad T = \begin{bmatrix} T_1^T \\ T_2^T \\ \vdots \\ T_N^T \end{bmatrix}_{N \times m}. \tag{13}$$

In order to train single hidden layer neural network, we hope to get $\widehat{\beta}_i, \widehat{w}_i \widehat{b}_i$, making

$$\left\| H\left(\widehat{w}_i, \widehat{b}_i\right) \widehat{\beta}_i - T \right\| = \min_{W,b,\beta} \left\| H\left(w_i, b_i\right) - T \right\|, \tag{14}$$

where $i = 1, 2, \ldots, L$, which is equivalent to minimizing the loss function:

$$E = \sum_{j=1}^{N} \left(\sum_{i=1}^{L} \beta_i g\left(W_i \cdot X_j + b_i\right) - t_j \right)^2. \tag{15}$$

The traditional learning algorithm based gradient needs to adjust all parameters in the process of iteration. In the ELM algorithm, once the input weight W_j and the bias b_i of the hidden layer are determined, the output matrix H of the hidden layer is uniquely determined [33]. Training single hidden layer neural network can be transformed into solving a linear system $H\beta = T$ and the output weight β can be determined as

$$\beta = H^+ T, \tag{16}$$

where H^+ is the Moore Penrose generalized inverse of the hidden layer output matrix, which is called pseudo inverse for short. Therefore, the learning process of ELM can be summarized into three points.

(1) Input weight and hidden layer threshold are given randomly

(2) According to the input of training data and the activation function of hidden layer, the output matrix H of hidden layer is calculated

(3) According to the formula (16), the output weight of the network is calculated

3. Model of Credit Risk Measurement of Small and Micro Enterprises

3.1. Construction of Credit Risk Measurement Index System. Because the construction of credit risk measurement indicators of small and micro enterprises has three characteristics of big data problem [9]: granularity scaling, cross-border association, and global view, which also belongs to the problem related to big data. First of all, granularity scaling refers to the digitalization of credit problem elements of small and micro enterprises, and it can be scaled among different measurement index granularity levels. Secondly, cross-border association refers to the expansion of the factor space of the credit problem of small and micro enterprises. It is not only necessary to consider the conventional elements and domain perspectives of small and micro enterprises, but also emphasize "externality" and "cross-border", which associate internal data (such as basic information and financial and nonfinancial and industry internal data of small and micro enterprises) with external data (social contact and e-commerce data of business owners or managers). Finally, the global view refers to the global nature of the definition and solution of the credit index problem of small and micro enterprises, emphasizing the control and interpretation of the overall picture of the relevant situation and its dynamic evolution.

Refering to the credit indicator system used by Moody's Investors Service and standard & Poor's corporation, combined with the development characteristics of small and micro enterprises driven by big data, we pay full attention to the impact of the social platform and e-commerce platform data of enterprise owners on the credit status of small and micro enterprises. Credit status of enterprise operators, enterprise innovation ability, enterprise competitiveness, and staff quality are highlighted, so as to reasonably reflect the credit level of small and micro enterprises. 7 first-level indicators and 22 second-level indicators are selected. SPSS software is used for factor analysis to quantify the correlation between indicators, and principal component analysis is used to extract factors from indicators. Finally, 7 primary indicators and 10 secondary indicators are selected as credit risk measurement indicator systems, which is shown as Table 1.

3.2. Model of Credit Risk Measurement

3.2.1. Ideas of the Integrated Algorithm. ELM can randomly initial the connection weights and hidden threshold between input layer and hidden layer before training. ELM does not

Table 1: Description of index system.

Index category	Specific index
Enterprise credit	x_1: sesame enterprise credit score
Development capacity	x_2: growth rate of total assets x_3: growth rate of owner's equity
Risk level	x_4: financial leverage
Profitability	x_5: net profit margin of total assets x_6: growth rate of return on net assets
Quality of enterprise owner	x_7: education level of enterprise owners
Operating capacity	x_8: sales expense growth rate x_9: sales expense rate
Innovation ability	x_{10}: number of patents

need iterative learning many times and can directly calculate the least square solution of output weight matrix. Although the learning speed is fast and the parameter adjustment is simple, the robustness of the model will be greatly affected when there is noise or uneven distribution in the training data set [34]. In addition, since the input weights and hidden layer thresholds are randomly selected, ELM needs more hidden layer nodes than other feedforward neural networks, which affect the stability and generalization of the network. Therefore, IGSO is used to optimize the weight and threshold of ELM, and the ensemble strategy is carried out for the best weights and thresholds and the most reasonable number of hidden layer nodes.

The idea of IGSO-ELM integrated learning algorithm is to determine the structure of ELM according to the input and output parameters, so as to determine the coding length of each individual glowworm. Each individual in the population contains the initial weights and thresholds value of ELM. That is to say, the initial weights and thresholds of ELM are obtained by decoding the glowworm individuals in the IGSO algorithm; then the IGSO algorithm is used to optimize the weights and thresholds of ELM. Thus, a parallel and interactive learning algorithm of IGSO and ELM method is built. Finally, the optimal ELM weights and thresholds are obtained. The flow chart of the IGSO-ELM is shown in Figure 6.

3.3. Description of the Integrated Algorithm. The number of ELM input nodes is determined by the credit risk measurement index of small and micro enterprises. The number of hidden layer nodes is decided by the number of samples. The Output node indicates whether the credit record of small and micro enterprises is in default. The IGSO-ELM algorithm is implemented as follows:

Step 1. Encoding: The parameters of ELM, such as weight W, threshold θ, and number of initial hidden layer nodes N, are considered as a glowworm to encode the real number. The population of glowworm is initialized by the theory of good point set. Calculate the output weights and create ELM network according to formulas (11)–(14) and (16).

Step 2. Initialing parameters: Set the size of glowworm population n, the initial fluorescein l_0, sensing radius r_s, initial step length s_1, s_2, fluorescein volatility coefficient ρ, and fluorescein renewal rate γ of each glowworm. Let the initial value of iteration counter $t = 0$. Set the maximum number of iterations iter max.

Step 3. Calculating glowworm fitness: Decode the glowworm, generate the weight and threshold of ELM, train and test the network to get the network test error (MAE), employ the error indices as the fitness of each glowworm, and update the fluorescein value of each glowworm by formula (1).

Step 4. Updating glowworm locations: Calculate the neighborhood set by formula (2), compute the moving probability by formula (3), and update the location of each glowworm i based on moving direction of target object selected by roulette method and step length calculated by formula (6).

Step 5. Updating Decision radius: Update the sensing radius r_d^i of glowworm i according to formula (5).

Step 6. Judging: Judge whether output accuracy of ELM meet the end conditions or not; if it is achieved, the optimistic results are given to the ELM network to produce the output result, iteration ends; otherwise, judge whether reaches the maximum iterations or not, if not, turn to Step 3. Otherwise, end.

3.4. Model Performance Evaluation Index. The performance evaluation criterion of the model is an indispensable part of the measure model. The proper estimation of the measure model can evaluate the accuracy of different models, which allow different models to compare with each other and also be used to define warning threshold [35]. There are many error measures method to evaluate the matching degree between the model and the observation data, such as MAE (mean absolute error), RMSE (root mean square error), CA (Classification Accuracy), and R^2 (square correlation coefficient). The details are as follows:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_D(i) - y(i)\right|, \tag{17}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_D(i) - y(i)\right)^2}, \tag{18}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y(i) - \overline{y}\right)^2}{\sum_{i=1}^{N}\left(y_D(i) - \overline{y}_D\right)^2}, \tag{19}$$

$$\mathrm{CA} = \frac{C_k}{T_n} \times 100\%, \tag{20}$$

where y_D and $\overline{y}_D$ represent the measure value and the average measure value; y and $\overline{y}$ represent the observed value and the average observed value; C_k and T_n represent the number of samples correctly classified and the number of test samples; N represents the number of observation samples.

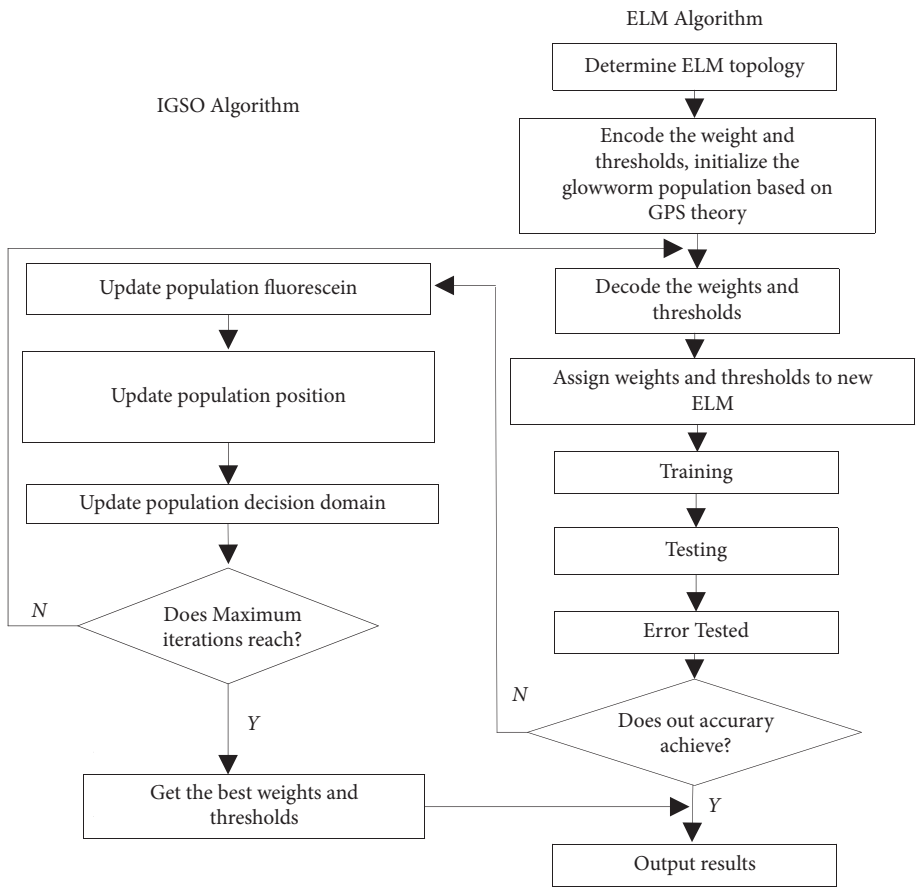


Figure 6: The flow chart of the IGSO-ELM.

4. Experiment and Analysis

All the codes in this experiment are written on MATLAB r2013a software platform. The compiled PC parameters are Intel (R) core (TM) i7-7200U CPU 2.71 GHz, 8.00 GB memory, 64-bit Windows10 operating system.

4.1. Performance of IGSO Algorithm Test Experiment. In order to verify the effectiveness of the IGSO algorithm based on good point sets theory and adaptive step length strategy, the following 10 benchmarks standard functions are selected for testing, as listed in Table 2. Among them, f_1 to f_5 are bidimensional functions and f_6 to f_{10} are multidimensional functions, with the dimension $d = 20$. To show the advantages of IGSO algorithm in solving multivalued functions, the study compared the IGSO algorithm with the traditional GSO algorithm is made as follows.

The parameters of the two GSO algorithms are setting as follows: the maximum iteration iter − max is set to 100. The population size n is set to 100. Set the initial fluorescein $l_0 = 5$, initial step length $s = 0.5$, $s_1 = 0.9999$, $s_2 = 0.0001$, fluorescein volatility coefficient $\rho = 0.4$, fluorescein renewal rate $\gamma = 0.6$, and field change rate $\beta = 0.08$ of each glowworm. The parameters of GSO and IGSO come from the value of experimental experience. The dynamic decision domain and perception radius of glowworm individuals have important influence on the results of the algorithm. Therefore, we focus on the experiment to determine the glowworm individual decision radius and perception radius when solving each Benchmark test function.

The initial perception domain and the dynamic decision domain for the test function $f_1, f_2, \ldots, f_{10}$ are different, whose decision radius of glowworm individuals are 2.448, 2.548, 2.688, 2.448, 2.888, 35.048, 28.048, 30.048, 35.048,30.048 respectively.

In Table 3, Best, Worst, Mean, and Var are used to record the best solution, the worst solution, the average solutions, and the variance of solutions for 30 times independent experiments. Figure 7 shows the convergence curve of the IGSO and GSO algorithms for 10 function $f_1, f_2, \ldots, f_{10}$. The performance test results of IGSO algorithm are analyzed as follows:

(1) In terms of calculation accuracy, in 30 times repeated experiments, Table 3 shows that the best value, the worst value, and the average value of 10 functions solved by the IGSO algorithm are all better than those of the traditional GSO algorithm, which are basically close to their respective standard values.

(2) In terms of convergence speed, it can be seen from Figure 7 that the convergence speed of the IGSO algorithm for functions f_1, f_5, f_9, and f_{10} is much faster than that of the GSO algorithm, but for functions f_6,

Table 2: Description of the 10 benchmark functions.

Name	Expression
Bohachecsky	$f_1 = x_1^2 + 2x_2^2 - 0.3 \times \cos(3\pi x_1) \times \cos(4\pi x_2) + 0.3$
Rosonbrock	$f_2 = 100 \times (x_1^2 - x_2^2) + (1 - x_1)^2$
Matyas	$f_3 = 0.26 \times (x_1^2 + x_2^2) - 0.482 x_1 x_2$
Booth	$f_4 = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
Ackley	$f_5 = (1.5 - x_1 + x_1 - x_2)^2 + (2.25 - x_1 + x_1 - x_2^2)^2 + (2.625 - x_1 + x_1 - x_2^3)^2$
Rastrigrin	$f_6 = \sum_{i=1}^{d} (x_i^2) - 10\cos(3\pi x_i)^2 + 10$
Dejong	$f_7 = \sum_{i=1}^{d-1} 100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2$
Zakharov	$f_8 = \sum_{i=1}^{d} (x_i^2) + (\sum_{i=1}^{d} 0.5 i x_i)^2 + (\sum_{i=1}^{d} 0.5 i x_i)^4$
Griewank	$f_9 = (1/4000) \sum_{i=1}^{d} (x^2) + \prod_{i=1}^{d} \cos(x_i / i^{(1/2)}) + 1$
Sphere	$f_{10} = \sum_{i=1}^{d} (x_i^2)$

Table 3: Results of 30 times independent experiments.

Function	Algorithm	Best	Worst	Mean	Var
f_1	GSO	$5.714551e-05$	$1.357698e-02$	$2.663643e-03$	$8.017839e-06$
	IGSO	$2.502637e-07$	$1.234684e-03$	$2.009736e-04$	$9.760807e-08$
f_2	GSO	$1.420992e-04$	$3.564882e-02$	$7.118286e-03$	$7.676224e-05$
	IGSO	$7.677080e-08$	$2.221346e-03$	$2.191989e-05$	$1.864490e-07$
f_3	GSO	$1.222893e-08$	$5.271113e-05$	$1.113159e-05$	$2.846101e-10$
	IGSO	$1.198294e-08$	$1.188994e-05$	$1.263033e-06$	$1.971471e-10$
f_4	GSO	$2.260599e-06$	$1.572854e-04$	$4.585418e-05$	$1.992108e-07$
	IGSO	$1.717953e-07$	$1.958495e-04$	$2.995911e-05$	$2.124073e-08$
f_5	GSO	$1.765632e-06$	$5.699369e-04$	$1.609170e-04$	$2.202052e-08$
	IGSO	$1.772661e-08$	$1.295454e-05$	$1.700449e-05$	$1.248972e-09$
f_6	GSO	$1.369385e-02$	$8.083180e-01$	$1.081428e-01$	$2.499092e-02$
	IGSO	$1.291843e-04$	$5.871926e-01$	$9.933543e-03$	$2.571551e-04$
f_7	GSO	$2.567627e-08$	$3.065898e-03$	$2.598793e-04$	$4.499324e-07$
	IGSO	$1.019086e-07$	$1.821743e-03$	$1.283270e-04$	$1.182948e-08$
f_8	GSO	$1.132974e-14$	$8.152062e-08$	$6.449133e-09$	$2.696393e-16$
	IGSO	$7.456504e-15$	$1.612441e-08$	$2.003430e-12$	$1.531690e-17$
f_9	GSO	$3.492894e-02$	$1.771344e-01$	$8.484216e-02$	$9.609253e-04$
	IGSO	$3.345529e-03$	$2.298949e-02$	$1.376466e-02$	$2.255257e-05$
f_{10}	GSO	$3.418968e-01$	$1.703526e+00$	$8.689655e-01$	$8.823356e-01$
	IGSO	$1.000030e-02$	$1.874430e+00$	$2.203945e-01$	$5.477851e-01$

f_7, and f_8 the convergence speed is the same as that of the GSO algorithm, which because functions f_6, f_7, and f_8 are multidimensional functions, and the search speed of two algorithms in high-dimensional space is much slower than that in two-dimensional space. The convergence speed of the IGSO algorithm for function f_2, f_3, and f_4 is much slower than that of the GSO algorithm. But the accuracy of the algorithm is better than that of the GSO algorithm.

(3) In terms of stability, from the variance value in Table 3, it can be seen that the GSO algorithm has a large difference in the 30 times solution results of function f_1, f_2, f_6, f_9, and f_{10}, and the variance of IGSO algorithm is smaller than that of GSO algorithm for function f_1 f_2, f_6, and f_9. This is because the initial population of IGSO algorithm is generated by the good point set method, and the initial solution is the same every time the algorithm runs. It can also be seen from Figure 7 that there is a certain fluctuation in the algorithm curve during the iteration process. Compared with IGSO algorithm, GSO algorithm is less stable.

Take function f_8 for example, comparing with the original GSO algorithm, the best solution, the worst solution, and the average solution of the 30 times independent experiments for IGSO algorithm decreased by 7.456504e−15, 1.612441e−08, and 2.003430e−12, respectively. The corresponding solution of original GSO algorithm decreased by 1.132974e−14, 1.612441e−08, and 6.449133e−09 respectively. The performance of the IGSO algorithm is the best, which indicates the abilities of IGSO algorithm to approximate to the most optimal solutions. Besides, the variance of the solutions for GSO and IGSO algorithm decreased by 2.696393e−16, 1.531690e−17 respectively, which means the IGSO algorithm can generate more accurate solutions stably. It is because the IGSO algorithm is based on the uniform distributed initial population produced by good point sets method and adaptive step-size strategy introduced by inertia weights, which made glowworm individuals in the algorithm keep an adaptive step-size to search for the optimal solution.

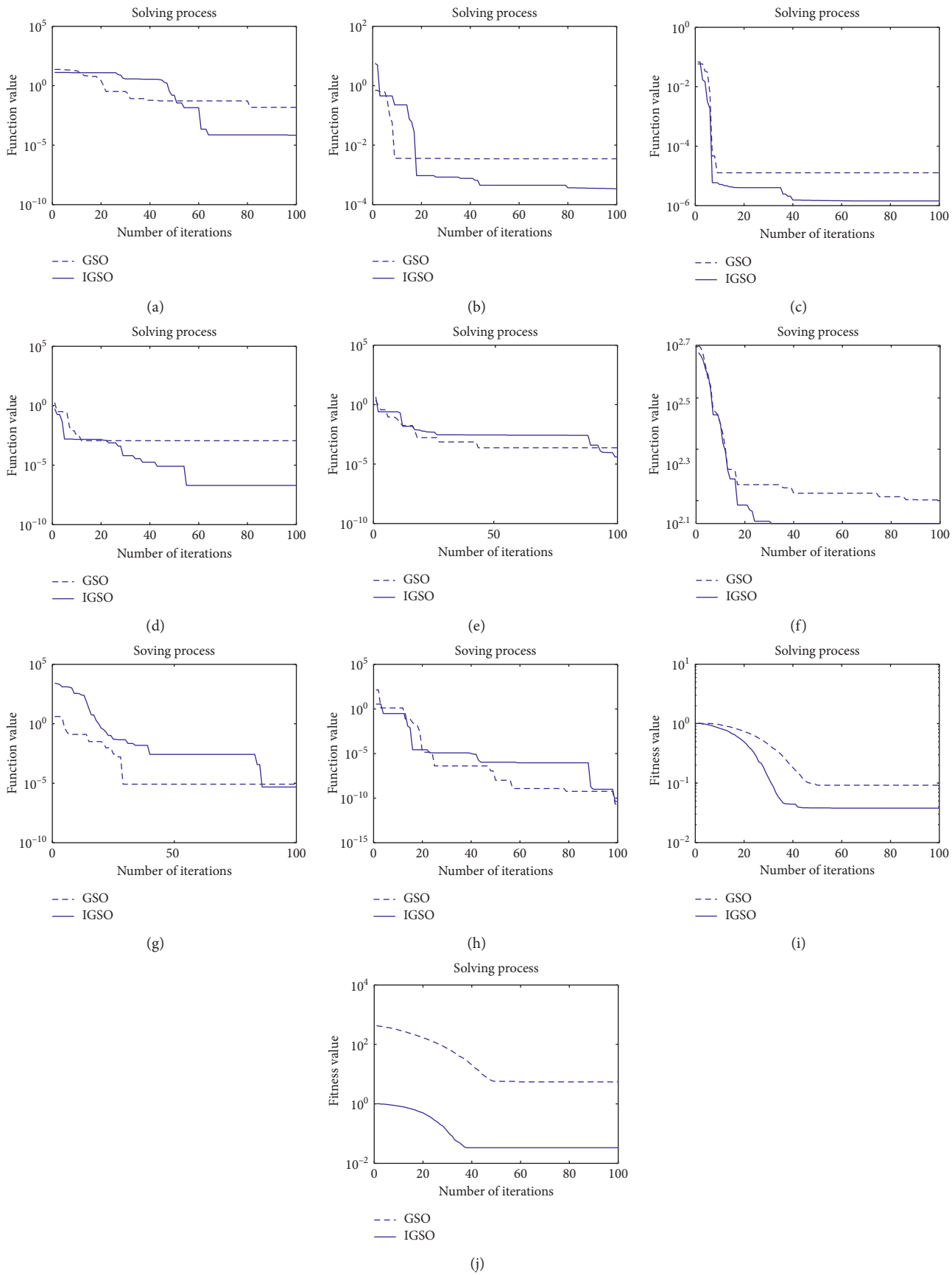


Figure 7: Convergence curves of the tested benchmark functions.

Therefore, the IGSO algorithm has dramatic global search ability.

In a word, compared with original the GSO algorithm, the IGSO algorithm has a great improvement in convergence speed and calculation accuracy, and the IGSO algorithm has some advantages in stability compared with GSO algorithm.

4.2. Empirical Analysis on Credit Risk of Small and Micro Enterprises

4.2.1. Data Collection. The credit data of small and micro enterprises in China are employed to test IGSO-ELM algorithm model. The experiment data span is from 2017 to 2018, which includes the following types of information: (1) personal information of legal person small and micro enterprises; (2) economic and financial ratio data of small and micro enterprises; and (3) current credit data of small and micro enterprises. Personal information of enterprise owners and financial and economic data are from CSMAR Taian Financial Research Database, and enterprise credit data are obtained from sesame credit business service platform. A total of 549 samples of small and micro enterprises have been obtained and processed. According to the definition of financial institutions, a loan is considered to be in default if it is overdue for more than 15 days. Among them, 308 small and micro enterprises did not have loan default, accounting for 56.11%, and the remaining 214 small and micro enterprises have different degrees of loan default, accounting for 43.89%. In order to effectively compare the classification models (BPNN, ELM, ABC-ELM, PSO-ELM, GSO-ELM, and IGSO-ELM), the data set is randomly divided into two disjoint subsets, of which 75% are training subsets and 25% are testing subsets. 10 cross tests are used for each model. The advantage of cross testing is that the credit model can contain the available data (75% of the samples) to the maximum extent.

4.2.2. Experimental Design. According to the research results in Section 4.1, 10 indexes are selected as the input layer of ELM neural network, the hidden layer of 10 nodes, and corresponding to the output layer of one node. So the ELM network structure is 10-10-1. To verify the efficiency of IGSO-ELM model, BPNN model, ELM model, ABC-ELM model, GSO-ELM model, and PSO-ELM model are constructed for credit risk measurement analysis. Parameters of ELM related models are set as follows: the reconstructed data are normalized between [0, 1]. The sigmoid function is employed as the activation function. The type of ELM is set to classification mode.

The parameters of model related to IGSO and GSO algorithms are set as the population size $n = 100$, the maximum iterative $\text{iter}\max = 50$. the initial fluorescein $l_0 = 5$, initial step length $s = 0.3$, $s_1 = 0.9999$, $s_2 = 0.0001$, fluorescein volatility coefficient $\rho = 0.4$, fluorescein renewal rate $\gamma = 0.6$, and field change rate $\beta = 0.08$ of each glowworm. The initial perception domain and the dynamic decision domain are the same, which are 58.248.

For BPNN model, the initial weights and thresholds are obtained by Marquardt Levenberg. The transfer functions of hidden layer and output layer are sigmoid function and linear function, respectively. Among them, the number of training iterations of BPNN model is 10^3, the MSE target is 10^{-2}, and learning rate is 10^{-1}. The number of hidden layer nodes in BPNN and ELM models is determined by step trial calculation. According to the size of training samples, the number of hidden layer nodes is increased in turn. The number of hidden layer nodes is determined when the classification accuracy reaches the maximum. The calculation results show that the number of hidden layer nodes of ELM model is 20.

For ABC-ELM algorithm [36], the parameters of model related to ABC algorithms are set as the population size n is 100, the number of employed foragers and onlookers are $n/2$, the limits of food sources Limit is set to 50, and the maximum iterative number is set to 50.

For PSO-ELM algorithm [37], the parameters of model related to PSO algorithm are set as the population size n is 100. The acceleration factors $C1$ and $C2$ are all set to 2, and other parameters of PSO algorithm are set as follows: maximum velocity $V_{\max} = 0.5$, minimum velocity $V_{\min} = -0.5$.

IGSO-ELM starts from the initial hidden node 5 and gradually increases the number of hidden nodes, which optimizes the sample classification accuracy. Determine the number of hidden nodes of IGSO-ELM as 20, which is shown as Figure 8. Thus, the structure of BPNN model, ELM model, ABC-ELM model, GSO-ELM model, and IGSO-ELM model are all 10-20-1.

In order to compare the convergence effect of IGSO-ELM model, GSO-ELM model, PSO-ELM model, and ABC-ELM model, Figure 9 shows the relationship between the fitness value, i.e. MSE (mean square error) and the number of iterations for four model. It can be seen from Figure 9 that the IGSO-ELM model reduces the number of iterations and can find a stable solution close to the best goal. The main reason is that the initial population based on the theory of good point set and the dynamic adjustment of the moving step length can improve the global search ability of glowworm population.

In Table 4, Best, Worst, Mean, and Var are used to record the best solution, the worst solution, the average solutions, and the variance solutions of BPNN model, ELM model, ABC-ELM mode, PSO-ELM mode, GSO-ELM mode, and IGSO-ELM mode for 10 times independent experiments of ABC-ELM, PSO-ELM, GSO-ELM, and IGSO-ELM algorithms. The test set output results of the six models are shown in Figure 10. The performance test results of six models are analyzed as follows.

Firstly, it can be seen from Table 4 that compared with the other five models, the best, worst, and average solutions of the IGSO-ELM model with 10 independent experiments are 92.7273, 85.4545, and 88.8951, respectively, which are better than the corresponding solutions of the other five models.

Secondly, compared with ABC-ELM model , PSO-ELM model , GSO-ELM model, and IGSO-ELM model, BPNN model and ELM model which are two kinds of single hidden layer feedforward neural network have lower classification accuracy. The best classification accuracy of BPNN model is just 50%, and the best classification accuracy of ELM model

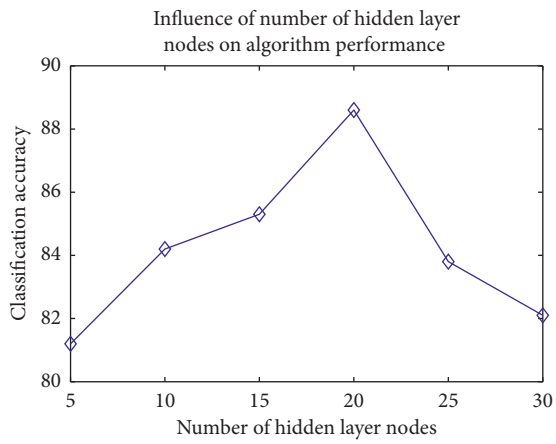


Figure 8: Prediction classification accuracy corresponding to increasing number of hidden nodes in IGSO-ELM.

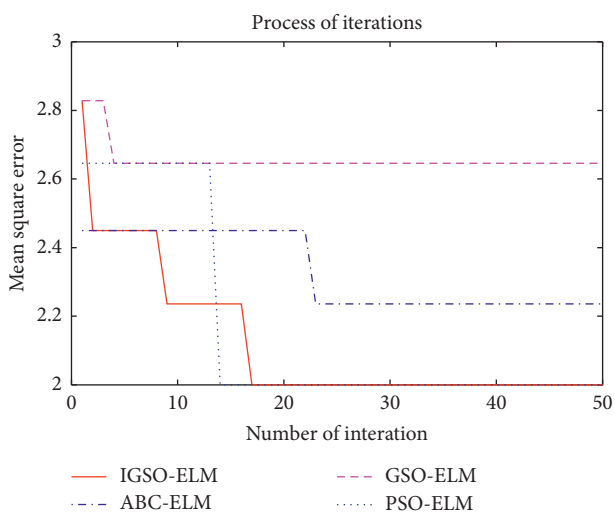


Figure 9: Relationship between MSE and the number of iterations.

Table 4: Classification Accuracy of 6 machine learning models.

Name	Best (%)	Worst (%)	Mean (%)	Var (%)
BPNN	50.9091	36.3636	40.6153	32.4702
ELM	81.8182	63.6364	74.2518	33.3884
ABC-ELM	89.0909	80.0000	83.9580	10.6115
PSO-ELM	90.9091	78.1818	87.2727	11.3264
GSO-ELM	90.9091	80.0000	85.3286	9.4113
IGSO-ELM	92.7273	85.4545	88.8951	6.7586

is only 81%, which shows that it is a very correct choice to optimize single hidden layer feedforward neural network model by using various swarm intelligence optimization algorithms to improve the classification accuracy.

Finally, it can be seen from Table 4 that the variance values of BPNN and ELM models are far greater than those of ABC-ELM, PSO-ELM, GSO-ELM, and IGSO-ELM models, which indicates that the stability of BPNN and ELM models is relatively poor compared with the other four models. The variance value of IGSO-ELM model is the smallest compared to those of ABC-ELM, PSO-ELM, GSO-ELM models, which indicates that IGSO-ELM model is more stable than the other three combination models. This is because the IGSO algorithm generates the initial population by using the good point set method, which can greatly improve the performance of the whole IGSO-ELM model.

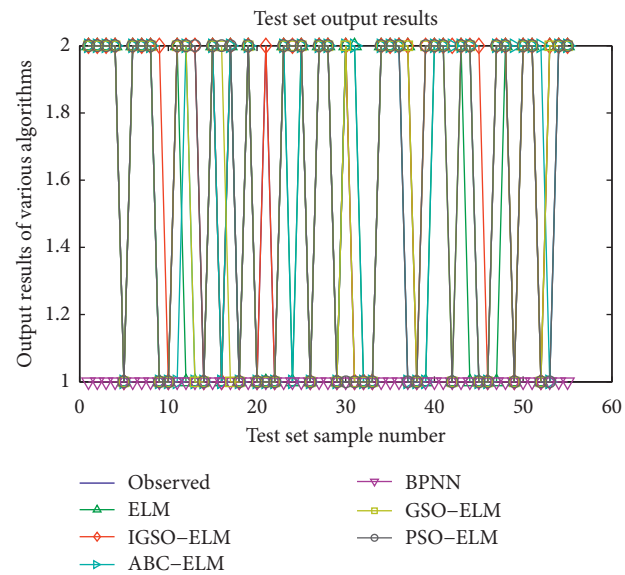


Figure 10: Test set output results of six algorithms.

In order to better illustrate the advantages of IGSO-ELM model, three commonly used evaluation indexes of machine learning, which are MAE, RMSE, and R^2, are selected. MAE (mean absolute error) is the mean absolute error, which means the average absolute error between the predicted value and the observed value; RMSE (root mean square error) is the sample standard deviation of the difference (called residual) between the predicted value and the observed value. RMSE is used to illustrate the degree of sample dispersion; R^2 is Nick name of the determination coefficient or the square correlation coefficients, which measures the fraction of the total variation in the dependent variable that is explained by the independent variable.

It can be seen from Table 5 that the square correlation coefficients R^2 of BPNN, ELM, ABC-ELM, PSO-ELM, GSO-ELM, and IGSO-ELM model are NaN, 0.3888, 0.5952, 0.6668, 0.6261, and 0.7105, respectively. The R^2 value of ELM model is larger than that of BPNN model, and the values of MAE and MSE are smaller than that of BPNN model, which shows ELM model has higher classification accuracy and better generalization performance compared with BPNN model based on gradient descent method.

In Table 5, the classification accuracy (CA) of ABC-ELM, PSO-ELM, and GSO-ELM models is higher than that of ELM model, which shows that swarm intelligent algorithm plays a good role in improving the prediction accuracy of ELM. In addition, the classification accuracy of IGSO-ELM model is higher than that of ABC-ELM, PSO-ELM, and GSO-ELM models, which shows the correctness of the improvement direction of ELM algorithm and the good performance of IGSO-ELM model in credit risk measurement of micro and small enterprise.

Table 5: Performance of 6 machine learning models.

Name	MAE	RMSE	R^2	CA
BPNN	0.4909	0.7006	NaN	50.9091
ELM	0.1818	0.4264	0.3686	81.8182
ABC-ELM	0.1091	0.3303	0.5952	89.0909
PSO-ELM	0.0909	0.3015	0.6668	90.9091
GSO-ELM	0.0909	0.3015	0.6261	90.9091
IGSO-ELM	0.0727	0.2697	0.7105	92.7273

5. Conclusion

Because of the lack of reliable financial statements and operating records, small and micro enterprises are facing financing difficulties, which has become an important factor restricting the development of small and micro enterprises. The credit status of small and micro enterprises plays an important role in their financing, so it is of great significance to study the credit risk measurement of small and micro enterprises. Therefore, a credit risk measurement model was proposed in this paper based on the improved GSO algorithm and ELM algorithm. Firstly, according to the growth and development characteristics of small and micro enterprises in the big data environment, the formation mechanism of credit risk of small and micro enterprises was analyzed from the perspective of granularity scale driven by big data, cross-border correlation, and global perspective, and a comprehensive evaluation index system was built by summarizing and analyzing the factors influencing credit evaluation indicators. Secondly, the traditional GSO algorithm was improved by good point set theory and variable step size strategy, and 10 benchmark standard functions were selected to test effectiveness of the IGSO algorithm. Experimental results show that the IGSO algorithm had great improvement in stability, accuracy, and convergence speed compared with the GSO algorithm. So, the integrated algorithm based on the improved GSO and ELM was established. The number of hidden layer nodes in ELM is determined by step-by-step trial method, and then the weight and threshold of ELM are optimized by the improved GSO algorithm in this integrated algorithm. Finally, ELM is a simple and effective method to establish the credit risk measurement model of small and micro enterprises which is verified by simulation experiment. Thus, a credit risk measurement model of small and micro enterprises based on IGSO-ELM integrated algorithm was proposed. The sample data of small and micro enterprises in China are collected, and the simulation experiment is carried out with MATLAB software tool. The experimental results showed that the model was effective, feasible, and accurate compared with the BPNN model, ELM model, ABC-ELM mode, PSO-ELM model, and GSO-ELM model. The research results of this paper can provide a reference for solving the problem of credit risk measurement of small and micro enterprises and also lay a solid foundation for the theoretical research of credit risk management.

Data Availability

The [.xlsx] data used to support the findings of this study are currently under embargo, while the research findings are commercialized. Requests for data, 6 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the fund of Philosophy and Social Science Planning Project of Anhui Province (No. AHSKY2018D09).

References

[1] X. Man, T. Zhang, C. Wang et al., "Credit risk factors identification and risk measurement of Micro , Small and medium enterprises in China," *Journal of Central University of Finance & Economics*, vol. 9, pp. 46–58, 2018.

[2] X. Han, "Study on the micro and small enterprises credit risks evaluation model based on support-vector machine regression ensemble," *Credit Reference*, vol. 37, no. 11, pp. 13–20, 2019.

[3] J. SU, "On soft information and credit risk evaluation of micro and small enterprises," *Credit Reference*, vol. 36, no. 7, pp. 13–20, 2018.

[4] F. Zhang, A. Li, and Y. Han, "Study on small and micro business credit assessment based on improved dynamic combined evaluation method," *Chinese Journal of Management*, vol. 16, no. 2, pp. 286–296, 2019.

[5] N. Yoshino and F. Taghizadeh-hesary, "Optimal credit guarantee ratio for small and medium-sized enterprises' financing: evidence from Asia," *Economic Analysis and Policy*, vol. 62, no. C, pp. 342–356, 2019.

[6] D. Hanggraeni, B. Ślusarczyk, L. A. K. Sulung, and A. Subroto, "The impact of internal, external and enterprise risk management on the performance of micro, small and medium enterprises," *Sustainability*, vol. 11, no. 7, pp. 2172–2189, 2019.

[7] L. Zhou, B. song, Y. Liu et al., "The application study of park enterprises credit risk evaluation based on euclid approach degree of fuzzy matter-element model," *Advances in Intelligent Systems Research*, vol. 143, pp. 279–283, 2018.

[8] B. Shi, G. Chi, and W. Li, "Exploring the mismatch between credit ratings and loss-given-default: a credit risk approach," *Economic Modelling*, vol. 85, pp. 420–428, 2020.

[9] G. Chen and G. Wu, "The challenges for big data driven research and applications in the context of managerial decision-making: paradigm shift and research directions," *Journal of Management Sciences in China*, vol. 21, no. 7, pp. 6–15, 2008.

[10] F. Shen, X. Zhao, Z. Li, Ke Li, and Z. Meng, "A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation"" *Physica A Statistical Mechanics & Its Applications*, vol. 526, 2019.

[11] A. Maher, F. Maysam, and Abbod, "A new hybrid ensemble credit scoring model based on classifiers consensus system approach"" *Expert Systems with Applications*, vol. 64, pp. 36–55, 2016.

[12] J. Abellán, J. G. Castellano, and G. Javier, "A comparative study on base classifiers in ensemble methods for credit scoring," *Expert Systems with Applications*, vol. 73, pp. 1–10, 2017.

[13] S.Y. Chang and T.Y. Yeh, ""An artificial immune classifier for credit scoring analysis"" *Applied Soft Computing*, vol. 12, no. No. 2, pp. 611–618, 2012.

[14] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: a novel physics-based algorithm," *Future Generation Computer Systems*, vol. 101, pp. 646–667, 2019.

[15] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, no. 8, pp. 849–872, 2019.

[16] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.

[17] K. N. Krishnanand and D. Ghose, "Glowworm swarm optimization algorithm for multimodal functions with collective robotics applications," *Multiagent and Grid Systems*, vol. 2, no. 3, pp. 209–222, 2006.

[18] K. N. Krishnanand and D. Ghose, "Glowworm swarm optimization: a new method for optimizing multi-modal functions," *International Journal of Computational Intelligence Studies*, vol. 1, no. 1, pp. 93–119, 2009.

[19] K. N. Krishnanand and D. Ghose, "Detection of multiple source locations using a glowworm metaphor with applications to collective robotics," in *Proceedings of 2005 IEEE Swarm Intelligence Symposium,*, pp. 84–91, Pasadena, CA, USA, June 2005.

[20] K. S. S. Rani and N. Devarajan, "Optimization model for sensor node deployment," *European Journal of Scientific Research*, vol. 70, no. 4, pp. 491–498, 2012.

[21] M. S. Couceiro, F. M. L. Martins, R. P. Rocha et al., "Mechanism and convergence analysis of a multi-robot swarm approach based on natural selection," *Journal of Intelligent & Robotic Systems*, vol. 76, no. 2, pp. 353–381, 2014.

[22] I. Aljarah and S. A. Ludwig, "A new clustering approach based on glowworm swarm optimization"" in *Proceedings of 2013 IEEE Congress on Evolutionary Computation*, IEEE, Cancun, Mexico, pp. 2642–2649, June 2013.

[23] H. Lo-keng and Y. Wang, *Applications of Number Theory to Numerical Analysis*, Springer, Berlin, Germany, 1972.

[24] Y. Chen, X. Liang, and Y. Huang, "Improved quantum particle swarm optimization based on good-point set," *Zhongnan Daxue Xuebao (Ziran Kexue Ban)/Journal of Central South University (Science and Technology)*, vol. 44, no. 4, pp. 1409–1414, 2013.

[25] J. Li, Z. Ni, X. Zhu et al., "Drought prediction model based on GPSGSO-BPNN parallel ensemble learning algorithm," *Systems Engineering Theory & Practice*, vol. 38, no. 5, pp. 1343–1353, 2018.

[26] L. Zhongqun, Z. Hua, and P. Chengming, ""Particle Swarm Optimization algorithm based on adaptive inertia weight'," in *Proceedings of 2010 2nd International Conference on. IEEE Signal Processing Systems (ICSPS)*, pp. 454–459, Dalian, China, July 2010.

[27] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of IEEE International Joint Conference on Neural Networks*, IEEE, Budapest, Hungary, July 2004.

[28] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 7, no. 1-3, pp. 489–501, 2006.

[29] A. Iosifidis, A. Tefas, and I. Pitas, "Approximate kernel extreme learning machine for large scale data classification," *Neurocomputing*, vol. 219, pp. 210–220, 2017.

[30] Y. Zhang, Y. Wang, J. Jin, B. Wang, X. Wang, and A. Cichocki, "Multi-kernel extreme learning machine for EEG classification in brain-computer interfaces," *Expert Systems with Applications*, vol. 96, pp. 302–310, 2018.

[31] Z. Chen, L. Wu, S. Cheng, P. Lin, Y. Wu, and W. Lin, "Intelligent fault diagnosis of photovoltaic arrays based on optimized kernel extreme learning machine and I-V characteristics," *Applied Energy*, vol. 204, pp. 912–931, 2017.

[32] Z. Huang, Y. Yu, and J. Gu, "An efficient method for traffic sign recognition based on extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 232–243, 2017.

[33] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: a review," *Neural Networks*, vol. 61, pp. 32–48, 2015.

[34] W.-Y. Deng, Q.-H. Zheng, X.-B. L. Xu et al., "Research on extreme learning of neural networks," *Chinese Journal of Computers*, vol. 33, no. 2, pp. 279–287, 2010.

[35] F. Salazar, R. Morán, M. Á Toledo et al., "Data-basedmodels for the prediction of dam behaviour: a review and some methodological considerations," *Archives of Computational Methods in Engineering*, vol. 42, no. 2, pp. 1–21, 2017.

[36] H. Yu, R. L. J. Ming, and Z. Shuping, "A hybrid model for financial time series forecasting-integration of EWT, ARIMA with the improved ABC optimized ELM," *IEEE Access*, vol. 8, pp. 84501–84518, 2020.

[37] Z. Tian, Y. Ren, and G. Wang, "Short-term wind speed prediction based on improved PSO algorithm optimized EM-ELM," *Energy Sources*, vol. 41, no. 6, pp. 26–46, 2019.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 5875803, 12 pages
https://doi.org/10.1155/2020/5875803

Research Article

Robustness Analysis of Urban Road Networks from Topological and Operational Perspectives

Wen-Long Shang,[1] **Yanyan Chen**,[1] **Chengcheng Song**,[1] **and Washington Y. Ochieng**[2]

[1]*Beijing Key Laboratory of Traffic Engineering, College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China*
[2]*Centre for Transport Studies, Imperial College London, London SW7 2AZ, UK*

Correspondence should be addressed to Wen-Long Shang; shangwl_imperial@bjut.edu.cn

Received 23 March 2020; Revised 12 June 2020; Accepted 22 June 2020; Published 14 August 2020

Guest Editor: Chunjia Han

This study comprehensively analyses the robustness of urban road networks through topological indices based on the complex network theory and operational indices based on traffic assignment theory: User Equilibrium (UE), System Optimum (SO), and Price of Anarchy (POA). Analysing topological indices may pin down the most important nodes for URNs from the perspective of connectivity, while more sophisticated operational indices are helpful to examine the importance of nodes for URNs by taking into account link capacity, travel demand, and drivers' behaviour. The previous way is calculated in a static way, which reduces the computation times and increases the efficiency for quick assessment of the robustness of URNs, while the latter is in a dynamic way, namely, calculating is based on removal of individual nodes, although this way is more likely to capture realistic meanings but consumes huge amount of time. The efforts made in this study try to find the relationship between topological and operational indices so as to assist the assessment of robustness of URNs to local disruptions. Seven realistic urban road networks such as Sioux Falls and Anaheim are used as network examples, and results show that different indices reflect robustness characteristics of urban road networks from different ways, and rank correlations between any two indices are poor although small network such as Sioux Falls have better correlations than others.

1. Introduction

Traffic road networks and infrastructure are frequently at risk of various types of natural disasters [1], such as the Indonesian tsunami and Hurricane Katrina. In addition, terrorisms are also increasingly rampant, for example, 9/11 attacks in 2001, London bombings in 2005, Paris attacks in 2015, and Brussels bombings in 2016 [2]. No matter what types these disasters are, all of these can cause huge destructions of our infrastructure. Therefore, the robustness of urban road networks has been a central focus of transport planners and managers.

The definitions of robustness of networked systems are presented in different fields. In the IEEE standard computer dictionary [3], robustness is defined as "*the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions.*" In the study of electrical networks, Holmgren [4] regards robustness as the ability to keep its structure intact when exposed to perturbations. Immers [5] regard robustness as "*the degree to which a system is capable of functioning according to its design specifications in the case of serious disruptions.*" Boccaletti et al. [6] define robustness as an ability of the network to avoid malfunctioning caused by the damage of a fraction of its constituents. Robustness is always associated with performance of systems [7].

To date, there are many indices and measures for the robustness of networked systems. Some of them measure from the perspective of network topology. Topological indices are all based on the topology of networks without taking into account the distribution of any quantity transported by the networks. These indices include degree

centrality, clustering coefficient, betweenness centrality, and the size of largest connected component, which are based on the complex network theory. Topological indices can be used to represent the ability the networked systems against failures or attacks due to the fact that these indices are usually related to the robustness of the network and some of these measures are able to reflect the efficiency and functionality of the underlying network [8–11], which are closely associated with the assessment of robustness. The definitions of these topological indices can be found in Section 2.

Due to the fact that the research related to the robustness assessment based on topological indices is very insufficient, here the studies of other networked systems concerning robustness are summarized. In order to explore the error and attack tolerance of the exponential and scale-free networks, Albert et al. [11] examine the change of diameter, the size of largest cluster, and the average size of the isolated clusters when networks suffer from random failure and targeted attacks. Following this, the network performance of complex networks, which are quantified by the average inverse geodesic length and the size of largest connected subgraph, is studied by Holme [8]. The result shows that removing nodes in a descending order of recalculated degree and betweenness of the current networks is more harmful than removals as the descending order of initial degree and betweenness. Similarly, Crucitti et al. [9] propose mathematical measures based on shortest paths to investigate the global and local efficiency of simulated networks. Afterwards, in order to investigate the cascading effects of electric networks, Crucitti et al. [12, 13] propose a measure of global efficiency to examine the performance of networks suffering from removal of nodes. In addition, Crucitti et al. [10] also examine the error and attack tolerance of transportation/communication networks using efficiency measure. To evaluate the dispersiveness/concentration of road networks in the presence of disasters, Sakakibara et al. [14] propose a topological index, and they define the most robust network as the one that minimizes the isolation of districts when suffering from catastrophic disasters.

Recently, topological analyses of air navigation networks for fifteen countries with a specific focus on betweenness have been conducted by Sun et al. [15]. They point out that a statistical distribution of betweenness of all the nodes in the network serves as an indicator of the network robustness because betweenness represents how frequently a given node lies in the shortest path between any pair of nodes, and more nodes with small betweenness implies that the network is robust against node failures.

The number of research studies on network robustness based on topological indices is considerable, but it is insufficient to capture capacity, drivers' behaviour, and flow patterns of realistic networks, especially for urban road networks.

Compared with the research of robustness from the topology perspective, operational indices, which take into account travel demand, drivers' behaviour, and link capacity, are also employed to access the robustness of networks. A new Network Robust Index (NRI) is proposed by Scott [16] to identify critical links of the network. The NRI is calculated based on the system wide, travel time cost of removing a given link and takes into account network flows, link capacity, and network topology. Following this, a network efficiency measure, which is based on travel disutility from the traffic user equilibrium model, is proposed by Nagurney and Qiang [17] to quantitatively assess the efficiency of congested networks and to rank the links of networks by capturing demand, flows, costs, and user behaviour. Based on this new measure, Nagurney and Qiang [18] utilize the relative change of network efficiency as an indicator to study the robustness of a transportation network while all link capacities decrease gradually. In addition, in order to assess the robustness of networks in the presence of link degradation, a relative total cost index is developed by Nagurney and Qiang [19]. This index is calculated based on total travel cost and is able to capture travel behaviour associated with User Optimization (UO) and System Optimization (SO). In the study, they quantify the robustness by gradually decreasing capacity of all links rather than removing links from the network.

The robustness measures based on operational indices is able to better capture more realistic characteristics of road networks such as travel demand, link capacity, and drivers' behaviour.

The above review suggests that systematic and comprehensive examination of different topological and operational indices is insufficient in the context of network robustness. It seems that the notion of robustness is used somewhat haphazardly. Although each study employs a reasonable network performance index for the certain application, these indices are never assessed on a similar platform and cross-compared with one another. This study aims to address this gap through a comprehensive and systematic comparison and analysis on the main robustness measures of different urban road networks from topological and operational perspectives when such networks are subject to local disruptions. Moreover, we will use Spearman's ranking correlation test to reveal potential correlations among topological indices and proposed operational indices.

The remaining of this paper is organized as follows. Section 2 introduces the methodologies used in this study, including topological indices, such as degree centrality, closeness centrality, and betweenness centrality, and proposed operational indices based on user equilibrium and system optimum principles. Afterwards, a case study including seven urban road networks is presented in Section 3. Conclusions and future works are shown in Section 4.

2. Methodology

The methodology used in the study is introduced in this section. The commonly used topological indices arising from the complex network theory, such as degree centrality, average path length, and betweenness centrality, are introduced in detail. Following this, operational indices based on user equilibrium (UE) and system optimum (SO), such as relative total UE and SO cost, are also presented to assess the robustness of URNs.

2.1. Topological Indices of Complex Network Theory

2.1.1. Degree Centrality. The degree centrality of a node i is the number of links connecting to this node [20] and is formulated as follows:

$$D_i = \sum_j b_{ij}, \tag{1}$$

where b_{ij} equals 1 if node i connecting with node j and equals 0 otherwise. This is a seemingly simple yet very important index, as it reflects the node's connectivity and its importance in its vicinity. The degree of a network is the average of the degree of all nodes. A larger network degree implies better connectivity within the network.

2.1.2. Clustering Coefficient. The clustering coefficient of a node i (CC_i) reflects the local feature and clustering effect of the network. It can be mathematically written as follows:

$$\mathrm{CC}_i = \frac{E_i}{k_i(k_i - 1)/2}, \tag{2}$$

where E_i is the actual number of edges between the neighbours of node i (a node connected with node i is called its neighbour) and k_i is the number of the neighbours of node i [21]. In social networks, this index tends to measure how close the friends of a given individual are.

2.1.3. Average Path Length. Average path length (APL) is the average length of all shortest paths between any pair of nodes, and it can be shown in the following equation:

$$\mathrm{APL} = \frac{1}{n(n-1)/2} \sum_{i>j} d_{ij}, \tag{3}$$

where d_{ij} is the length of the shortest path between i and j, if i and j are connected $d_{ij}$is 1 otherwise 0. APL, also named characteristic path length, can be employed to quantify the structural characteristics of networks [21]. The diameter of (D) of the network is the maximum $d_{ij}$for all node pairs. The APL and the D are usually to measure the efficiency and transport performance of the network.

2.1.4. Closeness Centrality. Closeness centrality [22] of a node i is the inverse of the sum of the shortest path distance from node $i$to other nodes and is normalized by $n - 1$, and it is written as follows:

$$\mathrm{C}_i = \frac{n-1}{\sum_{v_j \in v, i \neq j} d_{ij}}. \tag{4}$$

This index is used to measure the accessibility from the node i to other nodes.

2.1.5. Betweenness Centrality. Betweenness centrality (BC_i) is defined to examine the importance of each node i and can be mathematically expressed as follows:

$$\mathrm{BC}_i = \sum_{j,l \in V} \frac{N_{jl}(i)}{N_{jl}}, \tag{5}$$

where $N_{jl}(i)$ is the number of shortest paths passing through the node i and N_{jl} is the total number of shortest paths between any pair of nodes. The edge weight is the actual distance of the link so it is called weighted betweenness centrality.

2.1.6. Efficiency. This index is first proposed by Crucitti et al. [9] to explore the global efficiency of complex networks and looks very similar with APL, and its formulation is written as follows:

$$E(G) = \frac{1}{N(N-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}}. \tag{6}$$

The efficiency index (E) is already normalized, that is, $0 \leq E(G) \leq 1$, where G is a directed network. If the non-weighted network is completely connected, $E(G) = 1$. In this study, actual distance is assigned to weight of a link. The greater E, the better efficiency.

2.2. Operational Indices under Distinct Principles. In this section, we mainly introduce operational indices under user equilibrium and system optimum principles and also review some notations employed in this section. A list of notations used here is summarized in Table 1.

Here, h_p and f_a are different concepts, but they are closely related to each other. In the network, a link can be passed by one or more paths. Therefore, the link flow f_a is related to path flows and can be formulated as follows:

$$f_a = \sum_{p \in P} \delta_{ap} h_p, \quad \forall a \in A. \tag{7}$$

δ_{ap} is a link-path incident matrix:

$$\delta_{ap} = \begin{cases} 1, & \text{if } a \text{ belongs to } p, \\ 0, & \text{if } a \text{ does not belong to } p. \end{cases} \tag{8}$$

Path costs can be presented in the following way:

$$c_p = \sum_{a \in A} \delta_{ap} c_a(f), \quad \forall p \in P, \tag{9}$$

where $c_a$is a unit cost function depending on the link flow $f = (f_a\colon a \in A)$ and f_a denotes flows on each link $a \in A$. Here, we utilize Bureau of Public Roads [23] (BPR) link performance function as the link function:

$$t(f_a) = t_a\left(1 + \alpha\left(\frac{f_a}{K_a}\right)^{\beta}\right), \quad \forall a \in A, \tag{10}$$

where t_a and K_a are the free flow travel time and capacity of link a, respectively, and α and β are positive constants. Due to space limit, the values of all constants in BPR function for all URNs used in the study can be directed to https://github.com/bstabler/TransportationNetworks.

The following flow conservation must hold:

Table 1: List of notations used in Section 2.2.

Notation	Definition
$G(N, A)$	Represent a directed network, where N is a set of nodes and A is a set of links
W	Represent a set of origin-destination (OD) pairs
T_{os}	Represent a fixed travel demand for OD pair $(o, s) \in W$
P_{os}	Represent a set of paths connecting OD pair $(o, s) \in W$
c_p	Represent the unit travel cost over path $p \in P_{os}$
h_p	Represent the flow on path p
f_a	Represent the flow on link a
δ_{ap}	Represent a link-path incident matrix: Equal 1 if $a \in p$; equals 0 otherwise

$$T_{os} - \sum_{p \in P_{os}} h_p = 0, \quad \forall (o, s) \in W. \tag{11}$$

Then, the set of feasible flows is presented:

$$\gamma = \left\{ h \geq 0 \colon T_{os} - \sum_{p \in P_{os}} h_p = 0, \quad \forall (o, s) \in W \right\}. \tag{12}$$

2.2.1. Relative Total User Equilibrium (UE) Cost When Suffering from Local Disruptions. User Equilibrium (UE) is a principle proposed by Wardrop [24] and assumes that all individual drivers selfishly pursue personal interest maximisation. UE is also widely accepted as a variational inequality (VI) problem. The flow pattern $h^* = (h_p^* : p \in P)$ is a user equilibrium if and only if

$$\left. \begin{aligned} & h^* \in \Lambda, \\ & \sum_{p \in P} c_p \left(h_p - h_p^* \right) \geq 0, \quad \forall h \in \Omega, \end{aligned} \right\} \mathrm{VI}(c, \Omega). \tag{13}$$

The proof process can be found from [25].

In addition, the UE solution can be obtained from the following mathematical optimization program [26]:

$$\mathrm{Min}\, Z = \sum_a \int_0^{f_a} c_a(f_a) \mathrm{d} f, \tag{14}$$

which subjects to

$$\begin{aligned} & \sum_{p \in P_{os}} h_p = T_{os}, \forall (o, s) \in W, \\ & f_a = \sum_o \sum_s \sum_p \delta_{ap}^{os} h_p^{os}, \forall a \in A, \\ & h_p^{os} \geq 0, \forall p \in P_{os}, \forall (o, s) \in W, \\ & f_a \geq 0, \forall a \in A. \end{aligned} \tag{15}$$

In this case, the total cost is given by

$$\mathrm{TC} = \sum_{a \in A} f_a^* c_a(f_a^*). \tag{16}$$

$\mathrm{TC}_{\mathrm{UE}}$ is denoted as total cost when user equilibrium is reached and can be calculated based on equation (16). Following this, the relative total UE cost against local disruptions are denoted as $\mathrm{RTC}_{\mathrm{UE}}$. This new robustness index is extended based on Nagurney and Qiang [17] and Nagurney and Qiang [19]. $\mathrm{RTC}_{\mathrm{UE}}$ can be presented in the following way:

$$\mathrm{RTC}_{\mathrm{UE}}^i = \frac{\mathrm{TC}_{\mathrm{UE}}^i - \mathrm{TC}_{\mathrm{UE}}}{\mathrm{TC}_{\mathrm{UE}}} \times 100\%, \tag{17}$$

where $\mathrm{TC}_{\mathrm{UE}}^i$ is the total cost under UE principle when node i in the network is removed.

2.2.2. Relative Total System Optimum (SO) Cost When Suffering from Local Disruptions. The second principle of traffic assignment: System Optimum (SO), is also proposed by Wardrop [24]. In contrast to UE principle, SO assumes that there are central controllers to minimize the total cost of the whole road system rather than minimizing individual travellers' travel costs, and how drivers are assigned to routes so as to minimize the total travel costs is described by SO principle. The solution of SO is equivalent to the following nonlinear mathematical optimization program:

$$\mathrm{Min}\, Z = \sum_a f_a c_a(f_a), \tag{18}$$

which subjects to

$$\begin{aligned} & \sum_{p \in P_{os}} h_p = T_{os}, \forall (o, s) \in W, \\ & f_a = \sum_o \sum_s \sum_p \delta_{ap}^{os} h_p^{os}, \forall a \in A, \\ & h_p^{os} \geq 0, \forall p \in P_{os}, \forall (o, s) \in W, \\ & f_a \geq 0, \forall a \in A. \end{aligned} \tag{19}$$

In this case, $\mathrm{TC}_{\mathrm{SO}}$ is denoted as total cost when system optimum is reached and can be calculated based on equation (16).

Similarly, the relative total SO cost can be denoted as $\mathrm{RTC}_{\mathrm{SO}}^i$, which is also extended from Nagurney and Qiang [17]. $\mathrm{RTC}_{\mathrm{SO}}^i$ can be calculated by using equation (17):

$$\mathrm{RTC}_{\mathrm{SO}}^i = \frac{\mathrm{TC}_{\mathrm{SO}}^i - \mathrm{TC}_{\mathrm{SO}}}{\mathrm{TC}_{\mathrm{SO}}} \times 100\%, \tag{20}$$

where $\mathrm{TC}_{\mathrm{SO}}^i$ is the total cost under SO principle when node i in the network is removed.

2.2.3. Relative Price of Anarchy. Price of anarchy (POA) is proposed to measure the inefficiency of a system caused by selfish behaviour of individuals. The inefficiency of Nash Equilibria is an old idea [27], but Koutsoupias and Papadimitriou [28] first time proposed the term of POA

in the field of computer science. They utilised the ration between worst-case Nash equilibria to social optimum to measure the inefficiency of a system's lack of coordination [29]. In transport field, all drivers tend to selfishly choose routes so as to minimize their personal travel time, which causes the increase of total travel cost compared to system optimum [30]. In such context, POA is able to effectively quantify the inefficiency of transport networks with selfish routing [31], and it can be mathematically presented as follows:

$$\rho = \frac{\mathrm{TC}_{\mathrm{UE}}}{\mathrm{TC}_{\mathrm{SO}}}. \tag{21}$$

This ratio ρ captures the relationship between costs incurred user-oriented behaviour and system-oriented behaviour rule.

According to equation (21), relative price of anarchy (RPOA) can be formulated as follows:

$$\mathrm{RPOA}^i = \frac{\mathrm{TC}^i_{\mathrm{UE}}}{\mathrm{TC}^i_{\mathrm{SO}}} \times 100\%. \tag{22}$$

3. Case Study

In this study, for the first time, we systematically present the robustness analysis of URNs based on topological operational indices. In this section, Sioux Falls and Anaheim networks are depicted in detail, and other networks such as Tigarten centre, Mitte centre, and Prenzlauburg centre are analysed briefly due to space limit. The calculation of the solutions of UE and SO for large scale networks takes huge amount of computations; hence, only these seven networks are selected (all network data are obtained from https://github.com/bstabler/TransportationNetworks). In addition, here we also need to emphasize that local disruptions can be modelled well by removing one node at a time, and this study does not take into account removal of serval nodes at a time because the network performance fluctuates so greatly that the equilibrium is difficult to be achieved.

3.1. Sioux Falls Network. Sioux Falls is located in the east of South Dakoda and has a total area of 190.29 square kilometres [32]. The road networks of Sioux Falls are presented in Figure 1.

The Sioux Falls network consists of 24 nodes and 76 links. Assuming the network is subject to local disruptions, which can be simulated by removing nodes or links here. As equations (17) and (22), the relative total UE cost ($\mathrm{RTC}^i_{\mathrm{UE}}$), the relative total SO cost ($\mathrm{RTC}^i_{\mathrm{SO}}$), and the relative price of anarchy (RPOA^i) are calculated after individual nodes is removed. Detailed information is summarized in Table 2, and Figure 2 shows the total costs under UE and SO, as well as the POA when individual nodes are removed from the network.

As can be seen from Table 2, topological indices such as degree (D), weighted closeness (WC), clustering coefficient (CC), and weighted betweenness (WBC) centrality are included. These topological indices reflect a certain characteristic of network topology for URNs, which may be used to assess robustness of the networks. Here, Eff' denotes efficiency measure (see equation (6)) after individual nodes are removed, and link length in equation (6) is 1 if two nodes are connected otherwise 0; while WEff' represents weighted efficiency measure when individual nodes are removed, which regards realistic link length as the weight of the links. Both of these indices examine the performance of the network in terms of average path length. The 8th, 9th, and 10th columns are operational indices: $\mathrm{RTC}^i_{\mathrm{UE}}$ and $\mathrm{RTC}^i_{\mathrm{SO}}$ are total costs of UE and SO, respectively, when individual nodes are removed, and RPOA^i is the ratio of $\mathrm{RTC}^i_{\mathrm{UE}}$ to $\mathrm{RTC}^i_{\mathrm{SO}}$, which assesses the closeness of a UE network state from the system optimum state.

As can be seen from Figure 2, the $\mathrm{RTC}^i_{\mathrm{UE}}$ and $\mathrm{RTC}^i_{\mathrm{SO}}$ curves have similar tendency, and node #10 is the most important node in this network, since removing it brings the largest increase in the network-wide cost. Node #10 happens to have the largest degree in the network. The left graph of Figure 2 shows that node #10 is located in the central position of the network. However, the RPOA^i presents that node #2 impacts the network most greatly, though this node is located at the edge of the network.

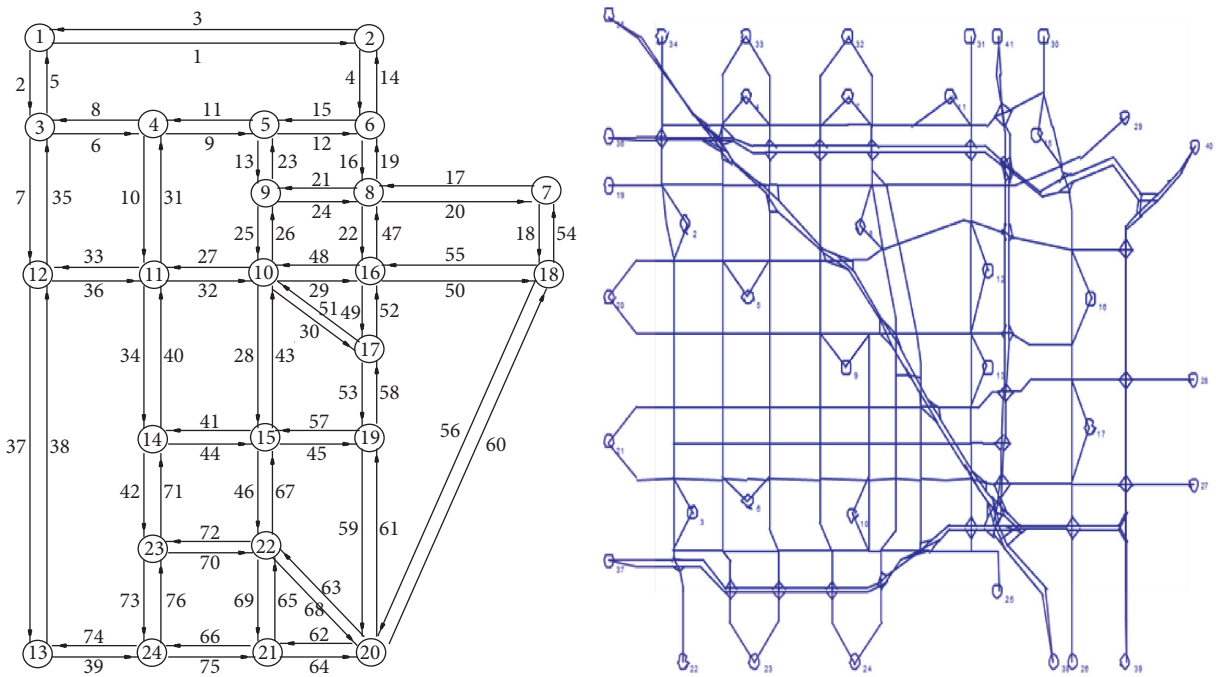
According to these quantitative topological and operational indices, we rank all the nodes in the network and the results, as shown in Table 3. Among all nine indices, Eff' and WEff' are ranked in an ascending order, which means nodes with lower efficiency values have the greatest impact on the network. As the rest of the indices, the nodes are ranked in a descending order, that is, nodes that can cause the greatest deterioration of relevant network performance are most important and also require more resources when such disruptions happen.

"—" means these nodes' ranking is meaningless due to the fact that these values are zero, which are caused when the neighbours of a given node in the network are not connected with each other.

Based on Tables 2 and 3, the relationship between those topological and operational indices are analysed as follows.

The spearman correlation method, also named ranking correlation, is used to explore such a relationship. Table 4 depicts the significance of correlation between each pair of indices.

As can be seen from Table 4, D has a high negative correlation with nonweighted efficiency index (Eff'), −0.8614, and also has relatively high correlations with RPOA^i, $\mathrm{RTC}^i_{\mathrm{UE}}$, and $\mathrm{RTC}^i_{\mathrm{SO}}$. *WC* has the highest correlation with D and weighed efficiency index, while the correlations between WC and operational indices are not very high. Due to the fact that many nodes have CC being 0, the correlations between CC and other topological indices are rather low. WBC is used to measure how frequently shortest paths pass through a given node, so weighted efficiency measure has the best correlation with it, and WBC has poor correlations with operational indices. Nonweighted efficiency index has low correlation with RPOA^i, $\mathrm{RTC}^i_{\mathrm{UE}}$, and $\mathrm{RTC}^i_{\mathrm{SO}}$, and weighted efficiency



(a) (b)

Figure 1: (a) Sioux Falls network. (b) Anaheim network.

Table 2: Topological indices and operational indices of all nodes for the Sioux Fall network.

Node	Indices								
	D	WC	CC	WBC	Eff'	WEff'	RTC^i_{UE}	RTC^i_{SO}	$RPOA^i$
1	0.0870	0.0667	0	0.0198	0.3965	0.1117	2.3588	2.4590	1.0088
2	0.0870	0.0691	0	0.0316	0.3961	0.1116	1.0941	1.1454	1.0140
3	0.1304	0.0799	0	0.1146	0.3878	0.1078	1.1437	1.2187	1.0038
4	0.1304	0.0895	0	0.1252	0.3871	0.1067	3.3681	3.5178	1.0044
5	0.1304	0.0920	0	0.1344	0.3888	0.1060	2.0273	2.1329	1.0039
6	0.1304	0.0906	0	0.1838	0.3863	0.1044	2.3281	2.4351	1.0065
7	0.0870	0.0909	0	0.0889	0.3955	0.1071	3.3825	3.5044	1.0107
8	0.1739	0.0927	0	0.1798	0.3814	0.1041	5.0474	5.2643	1.0029
9	0.1304	0.0898	0	0.0277	0.3878	0.1096	4.7399	4.9516	1.0019
10	0.2174	0.1018	0.0400	0.1067	0.3693	0.1062	14.3559	14.9324	1.0013
11	0.1739	0.0962	0	0.0982	0.3731	0.1085	6.7353	7.0059	1.0037
12	0.1304	0.0839	0	0.1120	0.3839	0.1080	4.1157	4.2802	1.0065
13	0.0870	0.0807	0	0.0817	0.3924	0.1086	4.0706	4.2363	1.0060
14	0.1304	0.0888	0	0.0567	0.3886	0.1091	4.2972	4.4696	1.0061
15	0.1739	0.0962	0	0.1344	0.3842	0.1060	6.3645	6.6235	1.0036
16	0.1739	0.1013	0.0667	0.1779	0.3821	0.1033	7.5824	7.9025	1.0015
17	0.1304	0.0970	0.1333	0.1028	0.3889	0.1053	11.7029	12.1898	1.0005
18	0.1304	0.0966	0	0.1304	0.3855	0.1053	1.8496	1.9364	1.0081
19	0.1304	0.0931	0	0.0968	0.3901	0.1056	3.6382	3.7820	1.0076
20	0.1739	0.0902	0.0652	0.0771	0.3809	0.1083	5.3172	5.5382	1.0037
21	0.1304	0.0885	0.1305	0.1014	0.3895	0.1076	3.1435	3.2669	1.0088
22	0.1739	0.0931	0.0652	0.1074	0.3874	0.1060	7.5057	7.8065	1.0034
23	0.1304	0.0858	0	0.0257	0.3917	0.1080	4.4205	4.6018	1.0052
24	0.1304	0.0855	0	0.1094	0.3887	0.1057	2.3496	2.4518	1.0081

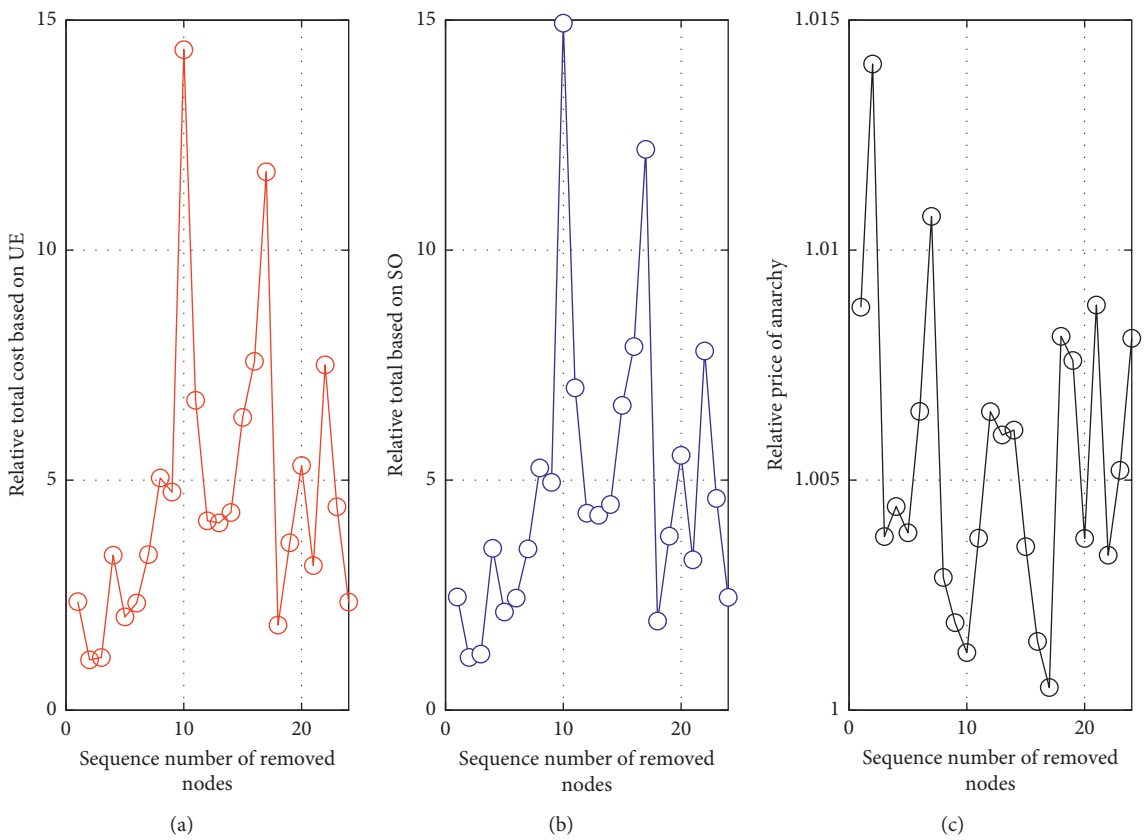


Figure 2: Evolution of RTC^i_{UE}, RTC^i_{SO}, and $RPOA^i$ of Sioux Falls when individual nodes are removed.

index is worse; the correlations between them are −0.1490, −0.1489, and −0.2906, respectively. Based on the above descriptions, we may see that only D index has relatively better correlations with the operational indices compared to the rest of topological indices. The explanation is that the larger the D of a given node implies more road links connected with the node, more likely the removal of the node is to give rise to apparent change of topology structure of the network, so as to cause significant fluctuations of network performance, which is directly related to the operational indices.

3.2. Anaheim Network. Anaheim is located in south east of Downtown Los Angeles, California. The right graph of Figure 1 presents the layout of the Anaheim road network.

As performed for the Sioux Falls network, nine topological and operational indices are calculated to indicate the network performance when suffering from local disruptions. Due to space limitation, only top 20 nodes of these indices are presented. As usual, with the exception of Eff' and WEff′, nodes are ranked in a descending order. Table 5 shows the top 20 nodes of these indices for Anaheim network.

Figure 3 depicts how the total UE cost, SO cost, and $RPOA^i$ change as certain node is removed from the network. As can be seen from the evolution of RTC^i_{UE} and RTC^i_{SO}, node #62 causes the most deterioration when removed, which highlights its importance, and $RPOA^i$ shows that node #195 has greatest impact on the price of anarchy. However, it seems that the rankings based on topological indices are not directly related to those based on operational indices.

In order to quantify the ranking correlations between these indices, the Spearman correlation method is employed. Table 6 presents the correlation coefficients between any pair of indices.

As can be seen from Table 6, RTC^i_{UE} and RTC^i_{SO} have very poor correlations with topological indices, even with degree, the correlations are just −0.4025 and −0.3819, respectively. $RPOA^i$ has no discernible correlations with these topological indices. In comparison, RTC^i_{UE} and RTC^i_{SO} have better correlations with degree, partially due to the homogeneity of the network, that is, nodes with more connections are more important in terms of both topological and operational perspective.

Due to the space limitation, the tables of node rankings and correlation between any pair of indices of other networks such as Tigarten centre, Mitte centre, and Prenzlauburg centre cannot be presented in the main body of this paper, and the detailed information can be shown in Supplementary Material.

Through analysis from ranking correlations of these URNs, there are still no significant correlations between topological and operational indices. However, the correlations between these indices are much better than

Table 3: Ranking of all nodes based on topological indices and operational indices for Sioux Falls.

Node	Indices								
	D	WC	CC	WBC	Eff'	WEff'	RTC^i_{UE}	RTC^i_{SO}	$RPOA^i$
1	21	24	—	24	10	16	18	18	3
2	22	23	—	21	11	8	24	24	1
3	8	22	—	8	20	6	23	23	15
4	9	15	—	7	8	17	16	15	13
5	10	10	—	4	16	18	21	21	14
6	11	12	—	1	12	19	20	20	8
7	23	11	—	17	15	24	15	16	2
8	2	9	—	2	18	5	8	8	20
9	12	14	—	22	6	15	9	9	21
10	1	1	6	12	4	22	1	1	23
11	3	5	—	15	22	10	5	5	16
12	13	20	—	9	3	4	12	12	9
13	24	21	—	18	9	7	13	13	11
14	14	16	—	20	14	21	11	11	10
15	4	6	—	5	24	3	6	6	18
16	5	2	3	3	5	12	3	3	22
17	15	3	1	13	17	23	2	2	24
18	16	4	—	6	21	20	22	22	5
19	17	7	—	16	19	11	14	14	7
20	6	13	4	19	23	13	7	7	17
21	18	17	2	14	13	14	17	17	4
22	7	8	5	11	7	9	4	4	19
23	19	18	—	23	2	2	10	10	12
24	20	19	—	10	1	1	19	19	6

'—' means these nodes' ranking is meaningless due to the fact that these 270 values are zero, which are caused by 271 that the neighbours of a given node in the network are not connected with each other.

Table 4: Spearman correlations between any indices pairs for Sioux Falls.

	D	WC	CC	WBC	Eff'	WEff'	RTC^i_{UE}	RTC^i_{SO}	$RPOA^i$
D	1	0.6762	0.4329	0.4376	−0.8614	−0.4187	0.6890	0.6972	−0.7336
WC	0.6762	1	0.4199	0.4505	−0.5608	−0.6612	0.6085	0.6050	−0.5910
CC	0.4329	0.4199	1	0.0469	−0.2110	−0.2654	0.5415	0.5415	−0.4341
WBC	0.4376	0.4505	0.0469	1	−0.5485	−0.8034	−0.0283	−0.0196	−0.2803
Eff'	−0.8614	−0.5608	−0.2110	−0.5485	1	0.3475	−0.4966	−0.5071	0.6054
WEff'	−0.4187	−0.6612	−0.2654	−0.8034	0.3475	1	−0.1489	−0.1498	0.2906
RTC^i_{UE}	0.6890	0.6085	0.5415	−0.0283	−0.4966	−0.1489	1	0.9991	−0.7746
RTC^i_{UE}	0.6972	0.6050	0.5415	−0.0196	−0.5071	−0.1498	0.9991	1	−0.7842
RTC^i_{UE}	−0.7336	−0.5910	−0.4341	−0.2803	0.6054	0.2906	−0.7746	−0.7842	1

those of Sioux Falls and Anaheim. In particular, the average weighted betweenness centrality (WBC) has the largest correlation with RTC^i_{UE} and RTC^i_{SO} among all topological indices used in this study, with values are uniformly above 0.63, and correlations for Mitte and MPF even up to 0.76 and 0.77 and 0.74 and 0.74, respectively. Therefore, for these networks, although the correlations are still not significant enough, which is expected, WBC seems to be relatively highly related to the total costs under UE and SO principles when networks suffer from local disruptions. In order to confirm this viewpoint, the normalized values of selected four indices for these networks are shown in Figure 4.

As can be seen from Figure 4, WBC has roughly consistent tendency with RTC^i_{UE} and RTC^i_{SO} when individual nodes are removed from these networks, though RTC^i_{UE} and RTC^i_{SO} in Tiergarten network are less fluctuated than WBC. While Eff' seems to have no consistent trend with other indices. The analysis based on Figure 4 confirms the previous viewpoint, that is, WBC has relatively higher correlations with total costs under UE and SO principles than other indices. A possible explanation is that the nodes that are frequently passed by shortest paths between all O-D pairs carry more flows, so the removal of the nodes with large WBC is more likely to cause the significant increase of total travel times of drivers.

Table 5: Top 20 nodes based on topological and operational indices for Anaheim.

Rank	Indices								
	D	WC	CC	WBC	Eff'	WEff'	RTC^i_{UE}	RTC^i_{SO}	$RPOA^i$
1	303	303	151	308	299	308	62	62	195
2	330	27	152	337	266	337	63	63	194
3	337	302	224	29	277	29	233	233	193
4	266	319	225	303	305	303	2	2	187
5	267	28	241	361	358	304	4	4	188
6	269	304	242	33	273	299	232	232	189
7	273	301	245	302	321	302	87	87	190
8	299	317	246	304	407	307	234	234	191
9	302	330	334	299	333	361	86	86	192
10	304	31	228	378	269	28	235	235	140
11	308	329	229	27	308	267	89	89	141
12	317	300	239	28	390	294	88	88	204
13	329	286	240	307	375	306	74	74	203
14	332	311	222	267	267	269	25	25	236
15	333	312	223	269	384	300	73	73	196
16	341	305	226	268	265	305	1	1	198
17	361	299	227	266	399	330	398	398	197
18	369	42	243	394	319	378	397	397	142
19	373	331	244	305	264	394	116	116	143
20	378	316	247	36	357	33	117	117	231

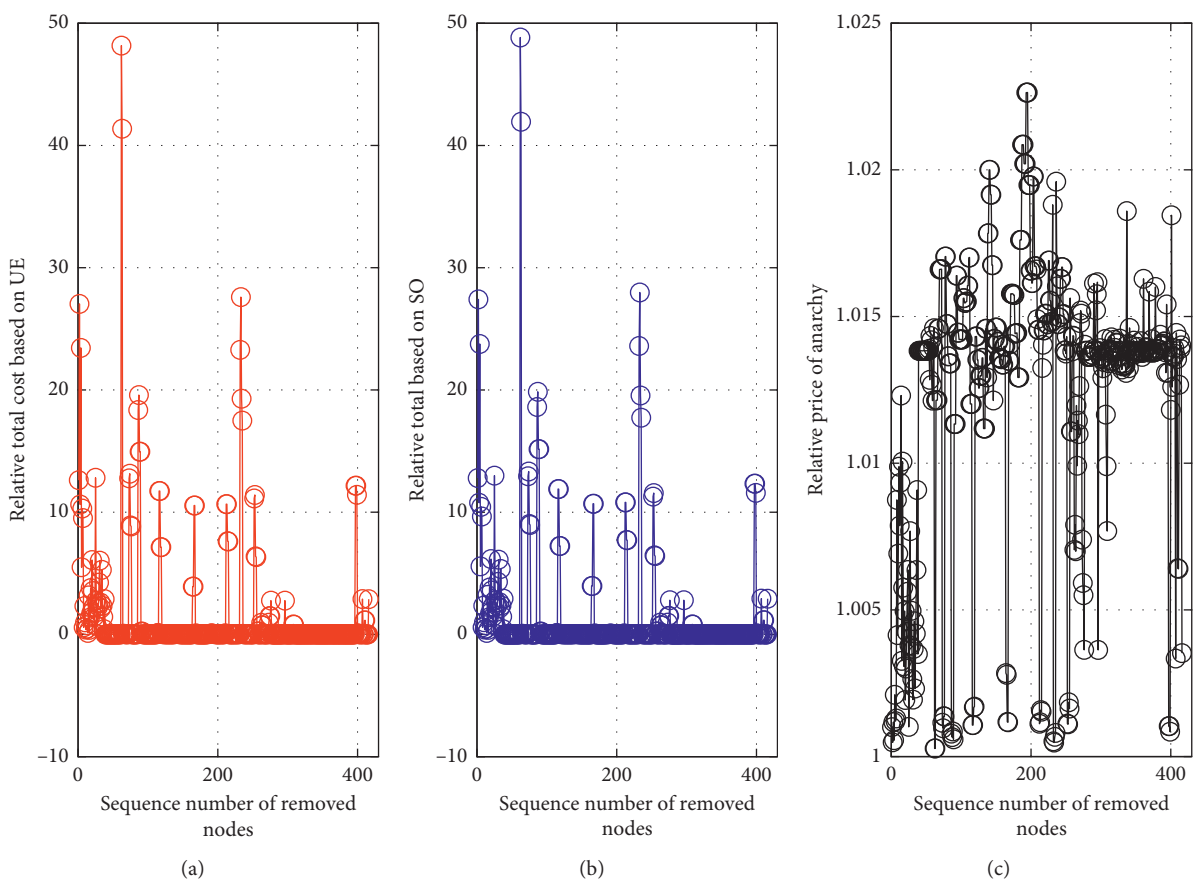


Figure 3: Evolution of RTC^i_{UE}, RTC^i_{SO}, and $RPOA^i$ of Anaheim when individual nodes are removed.

Table 6: Spearman correlations between any indices pairs for Anaheim.

	D	WC	CC	WBC	Eff′	WEff′	RTC^i_{UE}	RTC^i_{SO}	$RPOA^i$
D	1	0.3273	−0.1363	0.5477	−0.3227	−0.4445	−0.4025	−0.3819	−0.0272
WC	0.3273	1	0.0646	0.4808	−0.2002	−0.5073	−0.4737	−0.4766	0.1948
CC	−0.1363	0.0646	1	−0.0435	−0.0988	−0.0388	−0.0422	−0.0533	0.3189
WBC	0.5477	0.4808	−0.0435	1	−0.5208	−0.7688	−0.0703	−0.0341	−0.0267
Eff′	−0.3227	−0.2002	−0.0988	−0.5208	1	0.4698	−0.0965	−0.1008	0.0022
WEff′	−0.4445	−0.5073	−0.0388	−0.7688	0.4698	1	0.1025	0.0836	−0.1026
RTC^i_{UE}	−0.4025	−0.4737	−0.0422	−0.0703	−0.0965	0.1025	1	0.9917	−0.3260
RTC^i_{SO}	−0.3819	−0.4766	−0.0533	−0.0341	−0.1008	0.0836	0.9917	1	−0.3869
$RPOA^i$	−0.0272	0.1948	0.3189	−0.0267	0.0022	−0.1026	−0.3260	−0.3869	1

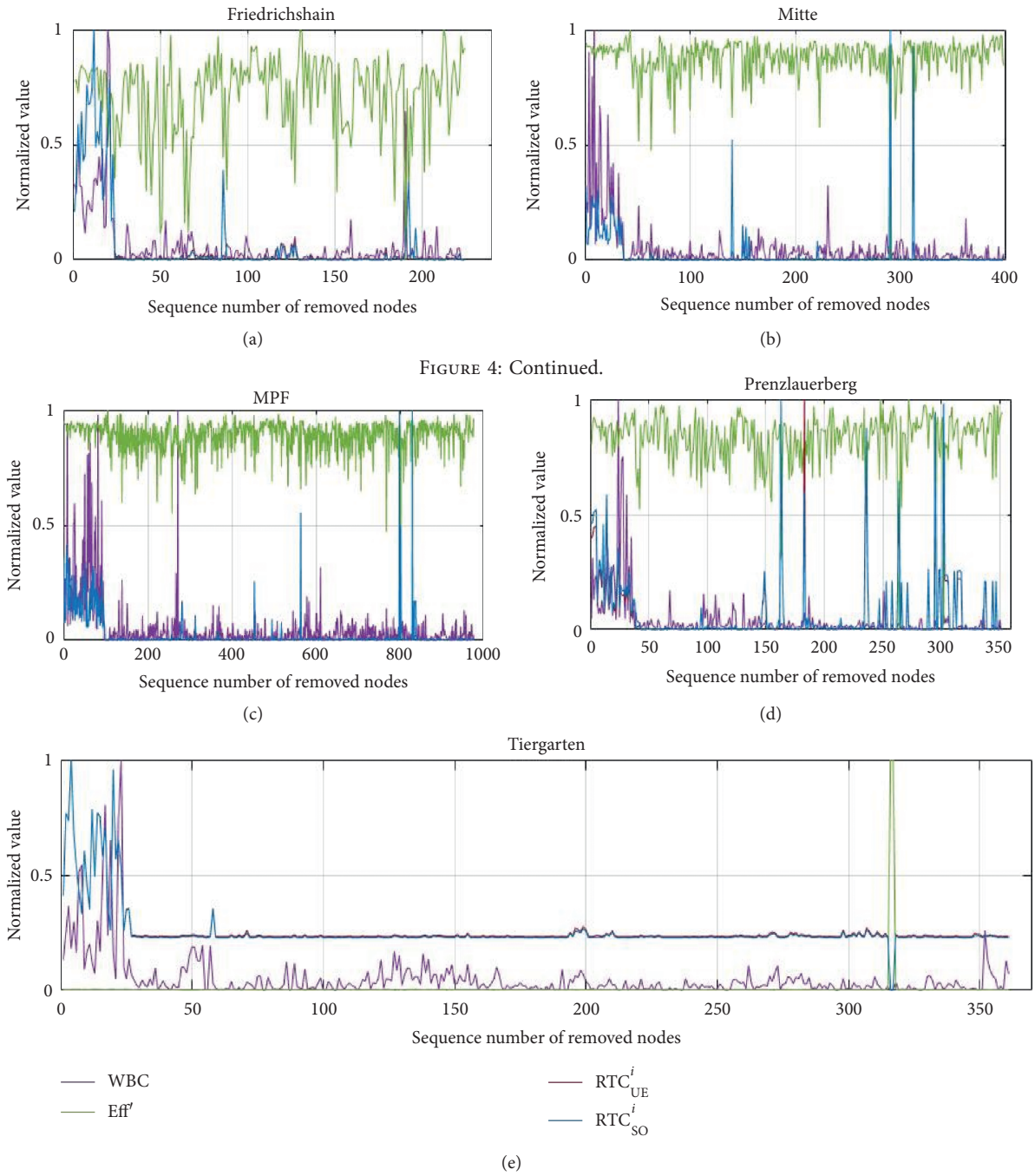


Figure 4: Continued.

Figure 4: Normalized values of WBC, Eff′, RTC^i_{UE}, and RTC^i_{SO} for (a) Friedrichshain, (b) Tigarten, (c) Mitte, (d) MPF, and (e) Prenzlauburg suffering from local disruptions.

4. Conclusions

To comprehensively analyse the robustness of URNs suffering from local disruptions, this study utilises topological indices and more sophisticated operational indices (i.e., those based on User Equilibrium (UE), System Optimum (SO), and Price of Anarchy (POA)) to examine the importance of nodes for selected URNs. The topological approach is comparatively more computationally efficient, while the latter takes into account more realistic characteristics and is calculated through certain iterative algorithms, which brings lots of computational overheads. The efforts to seek potential relationships between topological and operational indices have brought huge amount of valuable data for understanding the inherent structural characteristics of the networks:

(1) The results for the Sioux Falls network show that the node (#10) with the largest degree in the network is also the most important as assessed via user equilibrium ($\mathrm{RTC}^i_{\mathrm{UE}}$) and system optimum ($\mathrm{RTC}^i_{\mathrm{SO}}$) solutions. Moreover, degree has relatively high correlations with RPOA^i, $\mathrm{RTC}^i_{\mathrm{UE}}$, and $\mathrm{RTC}^i_{\mathrm{SO}}$ compared to other topological indices.

(2) The results for Anaheim show that a certain node (#62), when removed, causes the most deterioration in terms of $\mathrm{RTC}^i_{\mathrm{UE}}$ and $\mathrm{RTC}^i_{\mathrm{SO}}$ while node # 195 has greatest impact on the price of anarchy. $\mathrm{RTC}^i_{\mathrm{UE}}$ and $\mathrm{RTC}^i_{\mathrm{SO}}$ have poor correlations with topological indices, while the correlation between RPOA^i and topological indices is even poorer.

(3) The results for the rest of the selected URNs show higher correlations between topological and operational indices than Sioux Falls and Anaheim. In addition, weighted betweenness centrality (WBC) seems to be relatively highly correlated to the total costs under UE and SO principles when suffering from local disruptions.

Overall speaking, Spearman ranking correlations between topological and operational indices are not significant. However, degree and WBC stand out as the topological indices with relatively higher correlations with operational indices in some individual cases. This study seeks a rapid way of robustness assessment for managers and planners of urban transportation when suffering from local disruptions by exploring potential relationship between topological and operational indices. At this stage, the results of our study just shed light on the possibility and importance of such exploration. In the future, more advanced statistical/machine learning techniques such as deep learning and artificial neural networks can be utilised to provide a more profound understanding of the inherent structural connections of different network indices, thereby bridging the gap between complex network theory and traffic network modelling. In addition, this study only takes into account of removal of one individual node for the robustness assessment, and disequilibrium problems caused by the removal of multiple nodes in sequence may be explored in the future.

Data Availability

The data of urban road networks used in this study are available from database website https://github.com/bstabler/TransportationNetwork.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Chinese Scholarship Council (Grant 201308060082) and Key Special Project of Beijing City (Grant Z18110000391801).

Supplementary Materials

Figure 1: evolution of and of Friedrichshain when nodes are removed. Table 1: the values of top 30 nodes based on topological indices and operational indices for Friedrichshain. Table 2: top 30 nodes based on topological indices and operational indices for Friedrichshain. Table 3: Spearman correlations between any indices pairs for Friedrichshain. Figure 2: evolution of and of Mitte when nodes are removed. Table 4: the values of top 30 nodes based on topological indices and operational indices for Mitte. Table 5: top 30 nodes based on topological indices and operational indices for Mitte. Table 6: Spearman correlations between any indices pairs for Mitte. Figure 3: evolution of MFT when nodes are removed. Table 7: the values of top 30 nodes based on topological indices and operational indices for MPF. Table 8: top 30 nodes based on topological indices and operational indices for MPF. Table 9: Spearman correlations between any indices pairs for MPF. Figure 4: evolution of Prenzlauerberg when nodes are removed. Table 10: the values of top 30 nodes based on topological indices and operational indices for Prenzlauerberg. Table 11: top 30 nodes based on topological indices and operational indices for Prenzlauerberg. Table 12: Spearman correlations between any indices pairs for Prenzlauerberg. Figure 5: evolution of Tiergarten when nodes are removed. Table 13: top 30 nodes based on topological indices and operational indices for Tiergarten. Table 14: top 30 nodes based on topological indices and operational indices for Tiergarten. Table 15: Spearman correlations between any indices pairs for Tiergarten. (*Supplementary Materials*)

References

[1] A. Nagurney, "Building resilience into fragile transportation networks in an era of increasing disasters," in *Proceedings of the Transportation Research Board 90th Annual Meeting*, Washington, DC, USA, January 2011, https://supernet.isenberg.umass.edu/visuals/TRB_Panel_Nagurney_Talk.pdf.

[2] W. L. Shang, *Robustness and Resilience Analysis of Urban Road Networks*, PhD Thesis, Imperial College, London, UK, 2017.

[3] IEEE, IEEE Standard Computer Dictionary, *A Compilation of IEEE Standard Computer Glossaries*, Institute of Electrical and Electronical Engineers, Piscataway, NJ, USA, 1990.

[4] Å. Holmgren, "A framework for vulnerability assessment of electric power systems," *Critical Infrastructure Advances in Spatial Science*, vol. 3155 pages, 2007.
[5] B. Immers, "Reliability and robustness of transportation networks: problem survey and examples," in *Proceedings of the NECTAR Cluster Meeting on Reliability of Networks*, pp. 19-20, Amsterdam, The Netherlands, December 2004.
[6] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang, "Complex networks: structure and dynamics," *Physics Reports*, vol. 424, no. 4-5, pp. 175–308, 2006.
[7] M. Schillo, "Towards a definition of robustness for market-style open multi-agent systems," in *Proceedings of the Fifth International Conference on Autonomous Agents*, pp. 75-76, New York, NY, USA, June 2011.
[8] P. Holme, "Attack vulnerability of complex networks," *Physical Review E*, vol. 65, no. 5, 2002.
[9] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda, "Efficiency of scale-free networks: error and attack tolerance," *Physica A: Statistical Mechanics and Its Applications*, vol. 320, pp. 622–642, 2003.
[10] P. Crucitti, "Error and attack tolerance of complex networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 340, no. 1–3, pp. 388–394, 2004.
[11] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.
[12] P. Crucitti, V. Latora, and M. Marchiori, "Model for cascading failures in complex networks," *Physical Review E*, vol. 69, no. 4, 2004.
[13] P. Crucitti, V. Latora, and M. Marchiori, "A topological analysis of the Italian electric power grid," *Physica A: Statistical Mechanics and Its Applications*, vol. 338, no. 1-2, pp. 92–97, 2004.
[14] H. Sakakibara, Y. Kajitani, and N. Okada, "Road network robustness for avoiding functional isolation in disasters," *Journal of Transportation Engineering*, vol. 130, no. 5, pp. 560–567, 2004.
[15] X. Sun, S. Wandelt, and F. Linke, "Topological properties of the air navigation route system using complex network theory," in *Proceedings in the 6th International Conference on Research in Air Transportation*, Istanbu Technical University, Istanbul, TurkeyIstanbu Technical University, Istanbul, Turkey, May 2014.
[16] D. M. Scott, "Network robustness index-a new method for identifying critical links and evaluating the perfomance of transportation networks center for spatial analysis working paper," 2005.
[17] A. Nagurney and Q. Qiang, "A network efficiency measure for congested networks," *Europhysics Letters (EPL)*, vol. 79, no. 3, p. 38005, 2007.
[18] A. Nagurney and Q. Qiang, "Robustness of transportation networks subject to degradable links," *Europhysics Letters (EPL)*, vol. 80, no. 6, p. 68001, 2007.
[19] A. Nagurney and Q. Qiang, "A relative total cost index for the evaluation of transportation network robustness in the presence of degradable links and alternative travel behavior," *International Transactions in Operational Research*, vol. 16, no. 1, pp. 49–67, 2009.
[20] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
[21] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-"world" networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
[22] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, 1966.
[23] Bureau of Public Roads, *Traffic Assignment Manual*, U.S. Department of Commerce, Urban Planning Division, Washington, DC, USA, 1964.
[24] J. G. Wardrop, "Road paper. some theoretical aspects of road traffic research," *Proceedings of the Institution of Civil Engineers*, vol. 1, no. 3, pp. 325–362, 1952.
[25] T. L. Friesz, *Dynamic Optimization and Differential Games. International Series in Operations Research & Management Science*, p. 135, Springer US, New York, NY, USA, 2010.
[26] M. Beckmann, C. Mcguire, and C. Winsten, *Studies in the Economics of Transportation*, Yale University Press, New Haven, CT, USA, 1956, http://cowles.yale.edu/sites/default/files/files/pub/misc/specpub-beckmann-mcguire-winsten.pdf.
[27] P. Dubey, "Inefficiency of Nash equilibria," *Mathematics of Operations Research*, vol. 11, no. 1, pp. 1–8, 1986.
[28] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," in *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science*, pp. 404–413, Trier, Germany, March 1999.
[29] S. Robinson, "The price of anarchy," *SIAM News*, vol. 37, no. 5, 2004.
[30] H. Youn, M. T. Gastner, and H. Jeong, "Price of anarchy in transportation networks: efficiency and optimality control," *Physical Review Letters*, vol. 101, no. 12, pp. 1–4, 2008.
[31] T. Roughgarden, *Selfish Routing and the Price of Anarchy*, MIT Press, Cambridge, MA, USA, ISBN 0-262-18243-2, 2005.
[32] https://en.wikipedia.org/wiki/Sioux_Falls,South_Dakota.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1847517, 12 pages
https://doi.org/10.1155/2020/1847517

ResearchArticle

Relation and Application Method of Deep Learning Sea Target Detection and Segmentation Algorithm

Guangfu Li,[1,2] Zheng Wang,[2,3] and Jia Ren[2,4]

[1]*Department of Network and Information, Shandong University of Art & Design, Jinan 250300, Shandong, China*
[2]*Institute of Data Science, City University of Macau, Taipa, Macau, China*
[3]*School of Information Engineering, Shandong Youth University of Political Science, Jinan 250100, Shandong, China*
[4]*School of Information and Communication Engineering, Hainan University, Haikou 570228, Hainan, China*

Correspondence should be addressed to Jia Ren; renjiahnu@126.com

Received 24 April 2020; Revised 6 June 2020; Accepted 13 July 2020; Published 12 August 2020

Guest Editor: Chi-Hua Chen

Target detection and segmentation algorithms have long been one of the main research directions in the field of computer vision, especially in the study of sea surface image understanding, these two tasks often need to consider the collaborative work at the same time, which is very high for the computing processor performance requirements. This article aims to study the deep learning sea target detection and segmentation algorithm. This paper uses wavelet transform-based filtering method for speckle noise suppression, deep learning-based method for land masking, and the target detection part uses an improved CFAR cascade algorithm. Finally, the best separable features are selected to eliminate false alarms. In order to further illustrate the feasibility of the scheme, this paper uses measured data and simulation data to verify the scheme and discusses the effect of different signal-to-noise ratio, sea target type, and attitude on the algorithm performance. The research data show that the deep learning sea target detection and segmentation algorithm has good detection performance and is generally applicable to ship target detection of different types and different attitudes. The results show that the deep learning sea target detection and segmentation algorithm fully takes into account the irregular shape and texture of the interfering target detected in the optical remote sensing image so that the accuracy rate is 32.7% higher and the efficiency is increased by about 1.3 times. The deep learning sea target detection is compared with segmentation algorithm, and it has strong target characterization ability and can be applied to ship targets of different scales.

1. Introduction

Despite the rapid development of China's HNA industry, the current severe sea search and rescue conditions and environment have made the existing search and rescue system's emergency rescue capabilities facing severe tests. On the whole, China's existing emergency search technology equipment is still difficult to effectively meet the needs of the rapid development of maritime transportation, and its ability to deal with major disasters and accidents at sea is weak, and there is a large gap compared with the international advanced level. China's maritime search and rescue technology and equipment cannot meet the rapid search requirements for distress targets in the deep sea under harsh sea conditions. Therefore, in order to ensure the safety of people's lives and property and escort the emerging marine industry, we have a set of reliable and efficient maritime. It is very necessary to search for equipment to carry out the corresponding maritime search work.

Sea surface target detection is based on specific purposes and needs, by enhancing the image contrast method to enhance specific information in the image, while reducing or removing unimportant or unnecessary information, thereby improving the overall image quality. For the images obtained by different methods, through appropriate grayscale enhancement processing, the original image that can reduce the contrast or even cannot see any contour information at all is processed into a clear image that is convenient for

visual observation, making it suitable for subsequent target detection and identification [1]. At present, sea surface target detection methods are diverse and can be divided into two major categories: the first category is image enhancement methods based on the spatial domain; the second category is image enhancement methods based on the transform domain. In actual applications, depending on the specific application objects, occasions, and purposes, the methods used for sea target detection processing will also be different, and sometimes a combination of several methods will achieve better enhancement effects [2].

Satellite platform is an indispensable platform for marine target monitoring. Multisource information fusion can improve detection accuracy, recognition reliability, and space-time coverage and reduce the ambiguity of inference. In order to improve the ability of wide area search, accurate identification, continuous surveillance, and operational response of marine targets, it is necessary to further study the space-based marine target surveillance system and multi-satellite information fusion technology. He and Yao have studied the characteristics and development trend of information perception and fusion of marine environment and marine targets. They reviewed the data characteristics and shortcomings of the space-based maritime surveillance system and discussed the key technologies and future development directions in the fields of satellite networking, space-time fusion, and information fusion [3], but their methods were not forward-looking. The governments of the United States, Canada, and the European Union set up the Atlantic Ocean mapping International Working Group (asmiwg) according to the "Galway statement on Atlantic Cooperation," and made plans to map the entire Atlantic Ocean. The first step in this effort is to establish a target area of 400×400 km for the pilot mapping project. Here, scholars such as Wölfl et al. introduced the selection algorithm used to define these experimental areas. The algorithm they studied is based on careful selection of various stakeholders and publicly available marine environment parameters. Wölfl et al.'s research method involves a geographic information system-based coverage technology, which takes the parameters of the marine environment as a separate layer and combines them to obtain information about whether a location is suitable for the target area. Their research results reveal the suitability of the whole North Atlantic region and highlight three potential experimental mapping points [4]. The implementation process of this method is relatively complex, and the accuracy is not high.

The main problems that our country currently has in the technology of sea surface infrared image target detection based on the analysis of time and space characteristics available for research projects are as follows:

(1) Insufficient research on image preprocessing methods: at present, domestic scholars have not paid enough attention to image preprocessing when studying sea surface infrared image target detection algorithms. However, the special environment at sea determines that the infrared image cannot show high quality. For example, when detecting weak and small targets under high wind and wave conditions, strong sea wave interference will suppress or obscure the spatial characteristics of the target; under backlighting conditions, sea clutter affects the light strong reflection and refraction will seriously reduce the image quality, which will not only cause the image brightness distribution to be unbalanced but also destroy the original contour and contrast information of the target, which will bring great difficulties to the subsequent detection [5, 6]. Therefore, an excellent image preprocessing algorithm is extremely important for improving image quality, increasing the feature difference between the target and the background, improving the target detection rate, and improving the robustness of the algorithm.

(2) The target single frame detection algorithm has poor environmental adaptability. The infrared single-frame detection algorithm for sea targets currently available in China often has a relatively single application scenario or application target. Although the limitation of application scenarios or application goals is beneficial to extract a priori information, and thus improve the detection rate of the algorithm, however, in the actual sea target infrared search process, the scene information or target information cannot be predicted or limited. Therefore, the algorithm of the application condition list is difficult to be applied in actual engineering; otherwise, it will cause high leakage alarm rate or false alarm rate [7, 8]. In addition, in some harsh marine environments (such as dense fog, backlighting, and heavy wind), the existing single-frame detection algorithm is also difficult to accurately detect the target, so it cannot guarantee the robustness of the algorithm in different marine environments.

(3) The multiframe decision algorithm is seriously affected by the interframe jitter of the image sequence. In the actual sea target infrared search process, the imaging platform will be randomly shaken due to the influence of sea breezes or waves, which will cause the image sequence to appear "misaligned" between frames. This will destroy the continuity of the movement of the target between frames and the consistency of the trajectory, thereby adversely affecting the final target detection rate.

(4) The algorithm lacks systematic experimental verification. In the currently available literature or published research results, the experimental verification stage often uses a small amount of sea video images captured in specific scenarios to verify the algorithm and lacks verification data under complex and diverse environmental conditions, so it is difficult to reflect or guarantee. The environmental adaptability and robustness of the algorithm is extremely unfavorable for practical engineering applications [9, 10].

Due to the influence of bad weather such as foggy and rainy weather on the sea surface, long-distance transmission,

and atmospheric attenuation, the image after contrast stretching still has the problem of detail blur. In this paper, the wavelet transform-based filtering method is used for speckle noise suppression, and deep learning is used. The method is used for land masking, and the target detection part adopts the improved CFAR cascade algorithm and finally selects the best separable features for false alarm rejection. In order to further illustrate the feasibility of the scheme, this paper uses measured data and simulation data to verify the scheme and discusses the effect of different signal-to-noise ratio, sea target type, and attitude on the algorithm performance. The research data shows that the deep learning sea target detection and segmentation algorithm has good detection performance and is generally applicable to ship target detection of different types and different attitudes.

2. Proposed Method

2.1. Principle and Steps of Traditional Grabcut Image Target Segmentation Algorithm

2.1.1. Grabcut Image Segmentation. Grabcut image segmentation method is a graph-based segmentation method, which maps a graph into a network graph. By designing the energy function, the energy graph is segmented by the minimum cut, and the image is segmented by finding the minimum energy function. The color image is composed of pixels in the RGB color space [11, 12]. Because it is difficult to create a sufficient color space histogram, the GMM model (Gaussian mixture model) is used to build a color image data model. The basic steps are shown in Figure 1.

The current Grabcut algorithm uses an interactive method to manually mark the foreground target image and then complete the final segmentation operation according to the difference between the foreground and background. However, when processing a large number of images, manually marking the foreground target is a tedious and large amount of work with low efficiency [13, 14].

2.1.2. SSD Model Training. SSD convolutional neural network model training uses the open source TensorFlow deep learning framework provided by Google. There are 20 types of images in the PASCALVOC2007 database, including people, cars, airplanes, and various animals. During the SSD training process, you need to input the image and the real label box. After the convolution operation, the default boxes with different aspect ratios are evaluated at each position in the feature map with different scales. For each default box, it is predicted for all object categories, shape's offset, and confidence. During training, first match these default boxes to the real label boxes. As shown in Figure 2, the Jaccard similarity coefficient between the real target box and the default target box is calculated. The default target box with the largest similarity coefficient is paired with the real target box to ensure each real target frame has a default target frame corresponding to it [7, 15].

2.2. Characteristics of Sea Target Detection. In-depth study of sea surface infrared image target detection methods and the development of sea surface target detection mechanisms under complex sea conditions will help greatly improve China's maritime search and rescue capabilities, shorten the technical gap between China and developed countries, and thus provide life and property safety for the majority of marine personnel. Escort provides strong support for China's development to the world's major marine economy [16]. Target detection algorithm based on the analysis of space-time characteristics is the earliest and most classic infrared target detection algorithm [17, 18]. This kind of algorithm often uses a single frame detection and multiframe decision concatenation solution, first extract the suspect target area according to the spatial characteristics (such as gray contrast, texture, and gradient) in the single frame image, and then use the target to the continuity of motion and the consistency of the trajectory in the image sequence judge each suspected target, thereby eliminating the false target. At present, compared with the target detection algorithm based on the analysis of spatio-temporal characteristics, the target detection algorithm based on random signal processing can achieve more excellent target detection performance of sea surface infrared images. However, due to the extremely high-computational requirements of target detection algorithms based on marrow machine signal processing, even the use of program acceleration techniques (such as GPU and FPGA) cannot achieve real-time detection of targets [19, 20].

2.3. Features of Segmentation Algorithm. Image segmentation is an important research content of remote sensing image processing. The processing of visible light remote sensing images has a wide range of applications in military and civilian applications, and the segmentation of visible light remote sensing images is one of the important research contents [21]. Most of the existing remote sensing target extraction methods are based on a specific type of target on infrared images or SAR images, and the segmentation methods on visible light remote sensing images are mostly aimed at the classification of landforms on low and medium resolution images.

The feature classification is regarded as a kind of segmentation. Existing segmentation methods for such regional targets are usually inefficient and poor in real-time. For medium- and high-resolution object targets with clear boundaries, the current extraction method is basically in the semiautomatic processing stage of human experience interpretation or human-computer interaction and needs to solve the intelligence and automation of object extraction [21, 22].

In the infrared weak target detection algorithm designed in this paper, the feature extraction of the target adopts the method of extracting the minimum circumscribed rectangle of the target in the local neighborhood of the target, which is a very time-consuming method. In view of the fact that the system may be used in other occasions where the real-time requirement is higher or the target local neighborhood size is

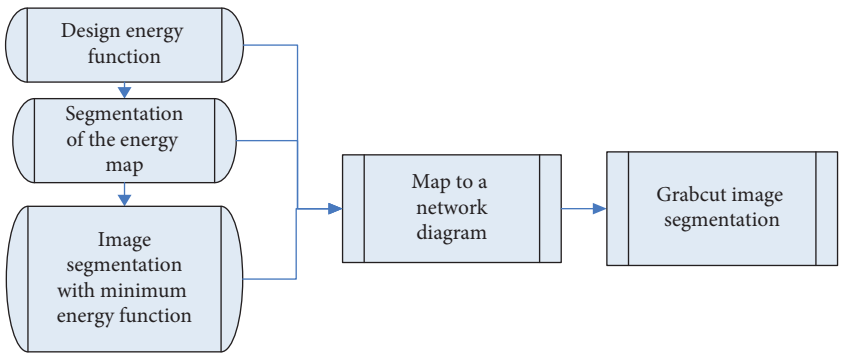


Figure 1: The basic steps.

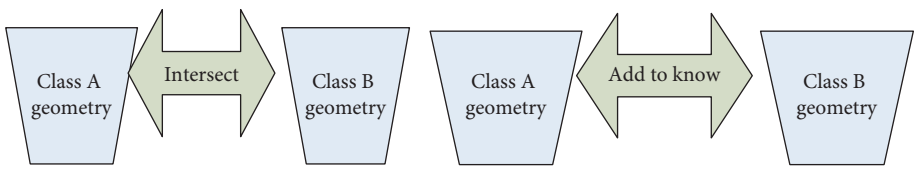


Figure 2: Jaccard similarity coefficient model.

larger, the method of circumscribed rectangles may not meet the real-time requirement, which requires the improvement of the target feature extraction method. Due to the problem of poor real-time target detection method based on convolutional neural network, in the next step, the CAFFE source code needs to be optimized and implemented, especially for the feedforward calculation of the CNN model, algorithm equivalent optimization and approximate optimization need to be considered, and performance is guaranteed under the premise of completing the hardware implementation of the prototype algorithm and through the laboratory injection simulation platform for testing and verification [23].

2.4. Grabcut Segmentation

2.4.1. Color Data Model. The Grabcut algorithm uses GM to establish a data model for color image target slices. It uses the full covariance GMM of K Gaussian components (generally $K = 5$) to model the foreground and background pixels of the image to obtain a Gaussian vector $k = \{k_1, \ldots, k_n, \ldots, k_N\}$, where k_n is the image. The nth pixel in the image belongs to the Gaussian component, k_n, $k_n \in [1, \ldots, K]$. Therefore, each pixel in the image corresponds to a Gaussian component from the target GMM or the background GMM, so the Gibbs energy of the entire image is

$$E = V + U. \tag{1}$$

In the formula, it is opacity, $A \in [0, 1]$, 0 represents the image background and 1 represents the foreground target; it is the statistical gray histogram between the background and foreground pixels of the image, $B = \{h(C, D), E = 0, 1\}$, z is an array of pixel values, and $h(\cdot)$ calculates gradient statistics. The influence of the main GMM component k introduces the GMM model:

$$U = \sum N + V. \tag{2}$$

In the formula, U is the regional energy term, representing the negative logarithm of the probability that the pixel belongs to the background or the target and indicating that a pixel in the image is classified as a background or target penalty. The mixed Gaussian probability density model is defined as follows:

$$D = \sum 1 + \sum i, \tag{3}$$

$$g = \frac{1}{\sum \exp}. \tag{4}$$

The negative logarithm of equation (3) can be written as follows:

$$d = -\log w + \frac{1}{2 \log k}. \tag{5}$$

In the formula, the parameters of the GMM have R three items: the weight of each Gaussian component, $A = B$ ((the number of pixels belonging to the Gaussian component)/(total number of pixels)); the average value of each Gaussian component μ, for the three channels of rgb. There are three element vectors, the covariance matrix $\sum$, which is a 3×3 matrix, expressed as follows:

$$\theta = w + a + k\infty. \tag{6}$$

Once the three parameters w, μ, and $\sum$ describing the background GMM and the target GMM of the image are determined, enter the RGB color values of the color image and bring it into the background GMM and the target GMM, and then the pixel can be obtained and classified into the background and target probability so that the U term of Gibbs energy can be determined.

The calculation formula of the boundary term energy V is as follows:

$$V = \sum \exp + \theta. \tag{7}$$

The boundary term reflects the penalty of discontinuity between neighboring pixels o and p. The greater the difference between adjacent pixels, the smaller the energy; the parameter β is determined by the pixel contrast; the Euclidean distance $|A|$ is used to measure the similarity between pixels. For images with low contrast, the difference between pixels o and $p|B|$ is relatively low, you need to multiply the larger β value to magnify the difference, otherwise multiply the smaller β value so that the energy of the V term can work normally; γ is a constant, and take $\gamma = 50$.

2.4.2. Iteratively Minimize Energy and Split

(1) Initialization:

(a) Frame the target in the image so that the pixels outside the box area are background pixels TB, and the pixels inside the box area are potential target pixels TU.

(b) For pixel $n \in TB$, let label $\alpha n = 0$, for pixel $n \in TU$, and let label $\alpha n = 1$.

(c) The GMM of the background and the target can be estimated through the foregoing steps, and the background and target pixels are clustered into K categories using the k-means method, that is, K Gaussian models of GMM, each of which corresponds to some pixel samples.

(2) Iterative minimization

(a) Assign Gaussian component K_n in GMM to each pixel:

$$K_n = rig \min. \tag{8}$$

(b) After the foregoing steps, each Gaussian model has some pixel samples. For the input image, learn the parameters w, μ of the optimization GMM optimization formula:

$$\theta = rig \min. \tag{9}$$

(c) Segmentation estimation: by minimizing Gibbs energy, the optimal formula of the initial segmentation algorithm is obtained:

$$\min = \min E. \tag{10}$$

(d) Repeating formulae (1)–(3) iteratively make the energy gradually decrease. After each iteration, the GMM and segmentation results are optimized interactively until the energy converges.

(e) Perform postprocessing such as smoothing on the boundary after image segmentation to complete the image segmentation process.

2.4.3. Algorithm Flow Description. The flow of this algorithm is described as follows. For an input image, the specific steps of the sea surface target detection and segmentation algorithm using this algorithm's deep learning are shown in Figures 3 and 4.

2.4.4. CFAR Algorithm Principle and Algorithm Model. The formation methods of the detection threshold of automatic detection are roughly: (a) formed by fixed threshold; (b) formed by the average amplitude of external interference; (c) formed on the basis of the obtained prior information of the interference statistical distribution; (d) formed prior to the interference-free statistical distribution information by testing under a statistical assumption of free distribution. Here, (a) is fixed threshold detection, (b) and (c) are adaptive threshold CFAR detection, and (d) is nonparametric CFAR detection. The clutter background of CFAR detection, which belongs to the category of automatic detection in a broad sense, can also be summarized into three typical situations: uniform clutter background; clutter edge; multitarget background. The Rayleigh distribution is suitable for the description of general clutter. The mean CFAR detection technology can achieve a constant false alarm rate. The mean class CFAR is suitable for a statistically stable background. It has a sliding window covering several distance units before and after the detection unit. The average of the reference samples in the sliding window is used to form a local estimate of the front and back edges. Local estimation average, large selection, small selection, or weighted average are to determine the average power estimation of the background clutter of the detection unit. In view of the fact that the signal may cross into the front and back neighboring units, the detection unit and its neighboring front and back distance units are generally not included in the average window. If the signal amplitude in the detection unit is greater than K times the average value in the sliding window, it is considered a signal. The algorithm model is shown in Figure 5. The echo sequence is sent to the transmission delay line. The central unit of the delay line is the current detection unit. There is a protection unit on each of the two adjacent sides. The front and back windows are summed separately and sent to the detection logic. If the detection logic outputs the average value of the sum of the front and rear windows, it is CA-CFAR; the selected small value of the front and rear window sum is SO-CFAR; the front and rear windows are output. The largest value of the sum is GO-CFAR.

3. Experiments

3.1. Dataset. The suppression of SAR speckle noise can be achieved by incoherent multiview processing, or it can be achieved by using spatial domain filtering. Incoherent multiview processing will reduce the ground resolution of the image. We can use spatial domain filtering methods, such as mean filtering, median filtering, Lee filtering, Kuan filtering, Frost filtering, Sigma filtering, and GammaMap filtering. However, this kind of algorithm has its own insurmountable contradictions: on the one hand, a larger filter

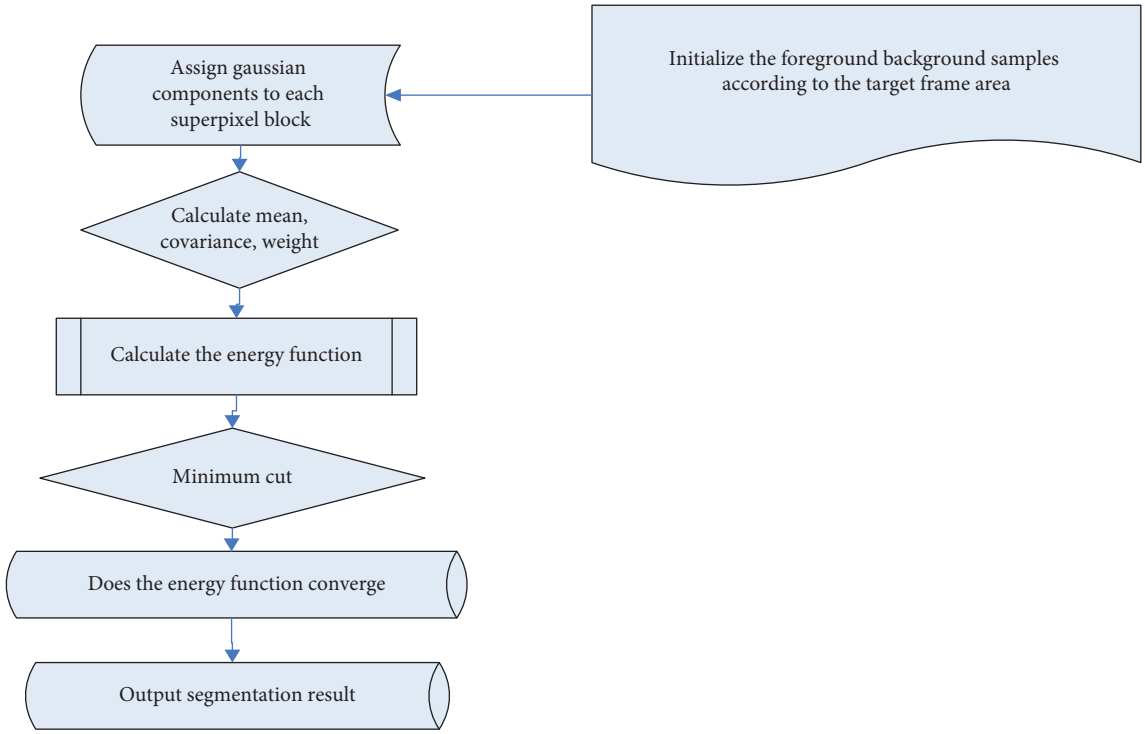


Figure 3: The specific steps of the detection and segmentation algorithm.

window needs to be selected to enhance the spot denoising effect, and on the other hand, the window selected must be small in order to maintain the actual resolution of the image. The wavelet transform filtering method can solve these problems, so in this paper, the wavelet transform filtering method of the sea surface image is used for speckle noise suppression, and the deep learning-based method is used for land masking. The target detection part uses an improved CFAR cascade algorithm, and finally selects the feature with the best separability is eliminated by false alarm. In order to further illustrate the feasibility of the scheme, this paper uses measured data and simulated data to verify the scheme and discusses the effects of different signal-to-noise ratios, sea target types, and attitude on the performance of the algorithm.

3.2. Basic Settings of the Experiment. In this paper, the wavelet transform filtering method of the sea surface image is used for speckle noise suppression, and the deep learning-based method is used for land masking. The target detection part uses an improved CFAR cascade algorithm and finally selects the best separable features for false alarm eliminate. In order to further illustrate the feasibility of the scheme, this paper uses measured data and simulation data to verify the scheme and discusses the effect of different signal-to-noise ratio, sea target type, and attitude on the algorithm performance.

3.3. Algorithm Description and Analysis. This algorithm has three main modules, and the specific content is shown below:

(1) Target detection operation: using deep learning network model to automatically detect the target rectangular frame in the image.

(2) Superpixel segmentation: calculate the objective function value through color information and position information. Update the cluster center according to the value of the objective function. By comparing the old and new cluster centers, if the cluster center changes, update the cluster information according to the new cluster center, then apply the visible light defogging algorithm to remove the infrared image blur, and finally get the super pixel segmentation result.

(3) Automatically mark the target frame as the Grabcut algorithm to initialize the sample information of the foreground and background according to the target detection results and process the sea surface defogging image through superpixelization, reducing the number of nodes in the graph, building a simple graph structure and reducing Grabcut the iteration time of the image segmentation algorithm.

3.4. Experimental Procedure. Use the deep learning target detection model to intelligently extract the target frame in the image instead of manually marking the foreground target in the Grabcut algorithm, and select the clustering center from the specification, and the boundary update strategy to improve the SLIC superpixel segmentation algorithm in two aspects, treating similar pixels as a super Pixels, construct a streamlined network graph, reduce the

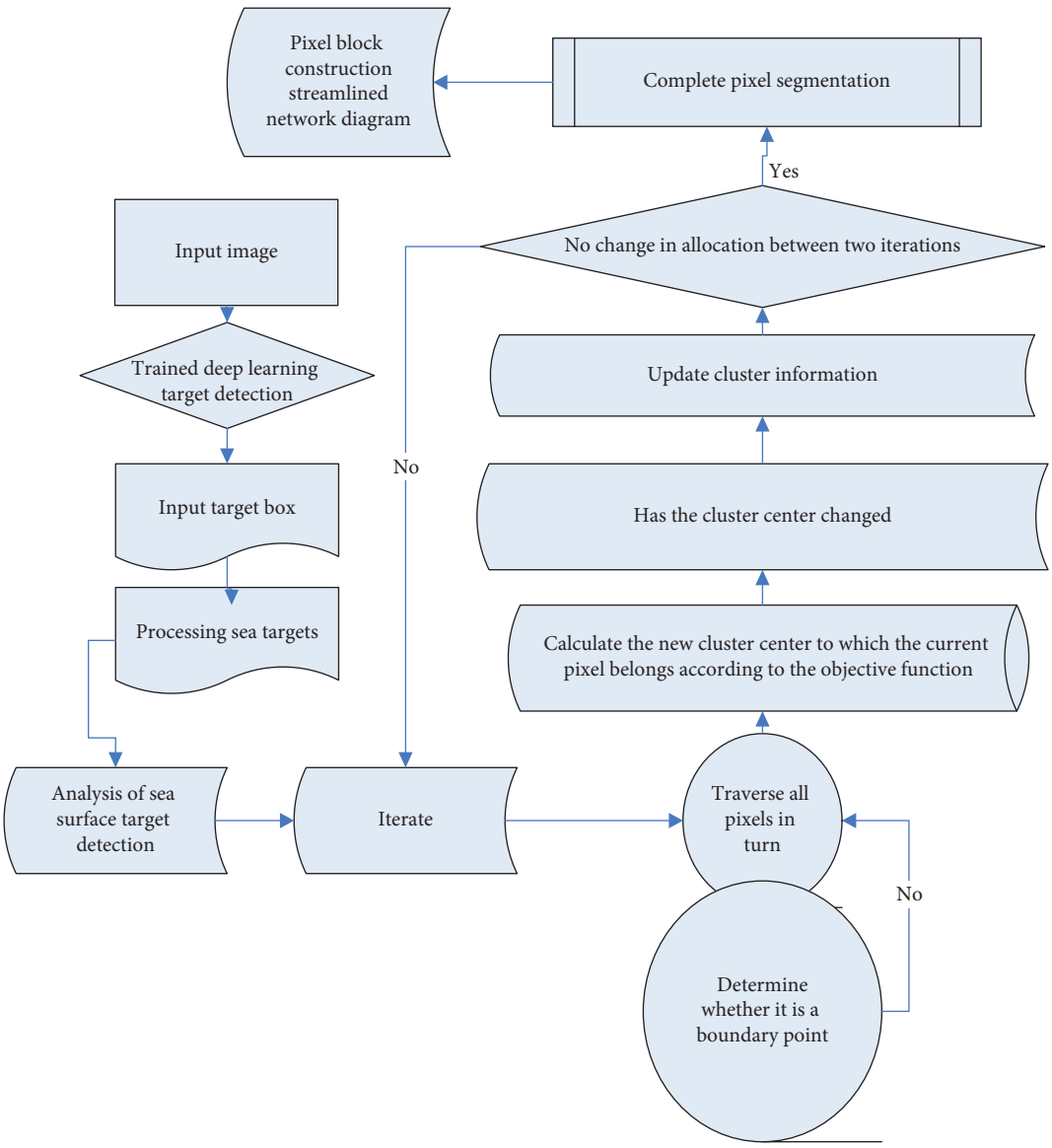


Figure 4: The specific steps of the detection and segmentation algorithm.

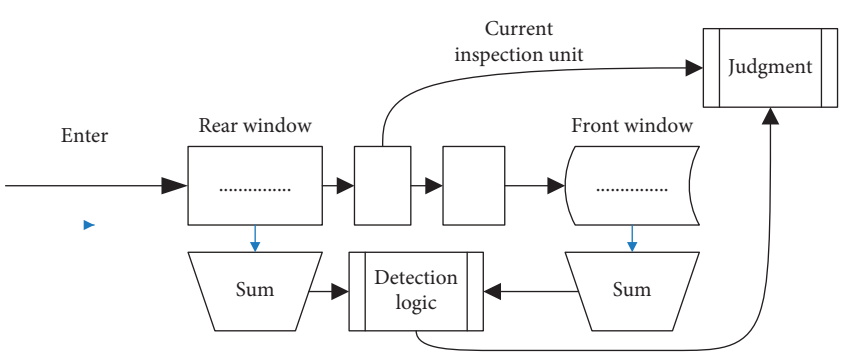


Figure 5: Algorithm model.

number of algorithm iterations, and improve the Grabcut segmentation algorithm from efficiency. Finally, to describe and analyze the steps of the improved Grabcut segmentation algorithm, iteratively select the minimum cut and finally complete the automatic target segmentation.

4. Discussion

4.1. Sea Surface Target Detection and Analysis

(1) As shown in Table 1 and Figure 6, the filtering method is used for speckle noise suppression, and the deep learning-based method is used for land masking. The target detection part uses an improved CFAR cascade algorithm, and finally the best separability is selected. False alarms are removed. The research found that the infrared imaging system has a long detection distance and the sea climate is complex and changeable. Due to the double impact of atmospheric scattering attenuation and bad weather on the sea surface, the attenuated transmitted light causes the reflected light received by the detector to reflect the essential information of the scene to change. After the contrast is enhanced, the infrared image of the sea target scene is still blurred; the target wheel, details, and other information are unclear. After the above measures have been taken after communication, the intensity contrast value of the original picture is more than 3 times higher than the screening value. The ratio will affect the subsequent target detection and recognition. The signal clutter must be deblurred after the contrast is enhanced. The blurred image after degraded seems to be covered with a layer of fog, which is similar to the effect of the fog compared to the visible light image. Inspired by this, we can use the visible light defogging algorithm to remove the infrared image blur.

(2) As shown in Table 2 and Figure 7, it is the clarity of the sea defogging image compared with the original under several measures. It can be seen from the several algorithms of the sea image defogging of the collected filtered data, the basic changes of the data after multilevel filtering, the use of side suppression filtering to change the appearance of the image, and the appearance after CAS filtering, and finally apply the original image defogging technology to collect relevant data, solve the image degradation problem caused by fog through these methods, and use the deep learning target detection model to intelligently extract the target frame in the image instead of manually marking the foreground target in the Grabcut algorithm, and separately, from the specifications, select the clustering center and use similar pixels as a superpixel to construct a simplified network graph, reduce the number of algorithm iterations and improve the Grabcut segmentation algorithm from the efficiency. Judging from the number of image segmentation, feature coefficients, and the number of marks, the quality after the image is fitted. It can be seen that the quality distribution coefficients of the percentages of multilevel filtering and GAS filtering in dealing with blurred image segmentation are almost equal, and the percentage of sharpness of the filtered image is lower than the original picture by about 50%. However, the results of feature visibility are not obvious. It shows that the sea surface image defogging process can increase the clarity and monitoring ability of the sea surface target and has a guiding significance for ocean detection and analysis.

4.2. Analysis of Sea Targets under the Algorithm

(1) The wavelet transform of the sea image is developed on the basis of two-dimensional discrete wavelet transform. Each time, through two-dimensional wavelet transform, we can get a low-frequency subimage and three high-frequency subimages in different directions. The low-frequency subgraph gives the main body contour information of the original image, while the high-frequency subgraph gives the original image edge details in all directions. After a wavelet transform, the obtained first-level low-frequency subgraph can be wavelet-transformed again to obtain a second-level subgraph, and the second-level low-frequency subgraph can still continue to undergo wavelet transform to obtain a third-level subgraph, and so on, as shown in Figure 8. In this schematic diagram, BB represents the low-frequency subgraph and BH, HB, and HH distributions represent the high-frequency subgraph in three different directions. The wavelet transform of the image is a pyramid structure, the bottom layer is the original image, and then each layer is decomposed upwards to obtain a layer of subimages, and the resolution of the image is gradually reduced from bottom to top. The original image is finally expressed as a combination of the top-level low-pass image and a series of prediction residual images of each layer image.

The basic idea of the speckle noise suppression method based on wavelet transform is to perform low-pass filtering on the low-frequency subgraph obtained after wavelet transform to filter out speckle noise, and then add the edge detail information contained in the three high-frequency subgraphs. It can achieve the dual purpose of not only removing speckle noise but also retaining the edge details of the image.

(2) The image sequence is an uninterrupted input when searching for sea targets. At this time, the target multiframe decision algorithm of the deep learning sea target detection and segmentation algorithm proposed in this paper can well achieve the removal of pseudotargets and background interference. As shown in Table 3 and Figure 9, select images are collected at different time periods in a certain sea area for analysis. First, the dark primary color a priori theory is used to estimate the transmission image, and the sharpness of the image is tested by a professional measuring instrument. The transmission diagram reflects the degree of the image affected

Table 1: Percentage of clarity of pictures in various situations.

	Filtering	Multistage filtering	Side suppression filtering	CAS filtering	Original image
Contrast	30	48	69	85	113
Signal to clutter	15	17	9	11	8
Scroll	65	68	94	58	74

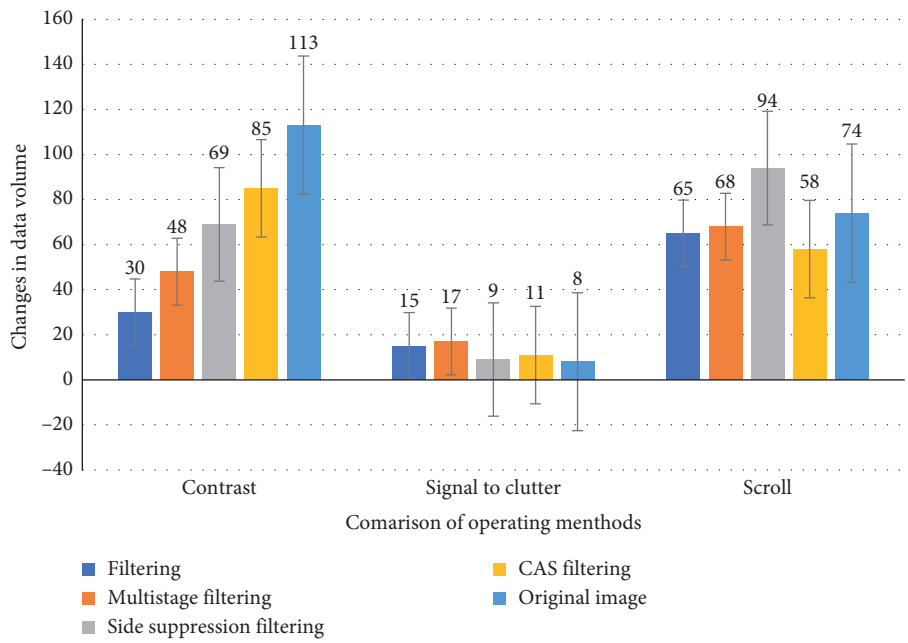


Figure 6: Percentage of clarity of pictures in various situations.

Table 2: Percentage of sea fog image clarity under the algorithm.

	Filtering	Multistage filtering	Side suppression filtering	CAS filtering	Original image
Image segmentation	27.3	29.8	26.4	29.7	20.8
Characteristic coefficient	45	34	35	46	46
Mark	245	643	543	564	453

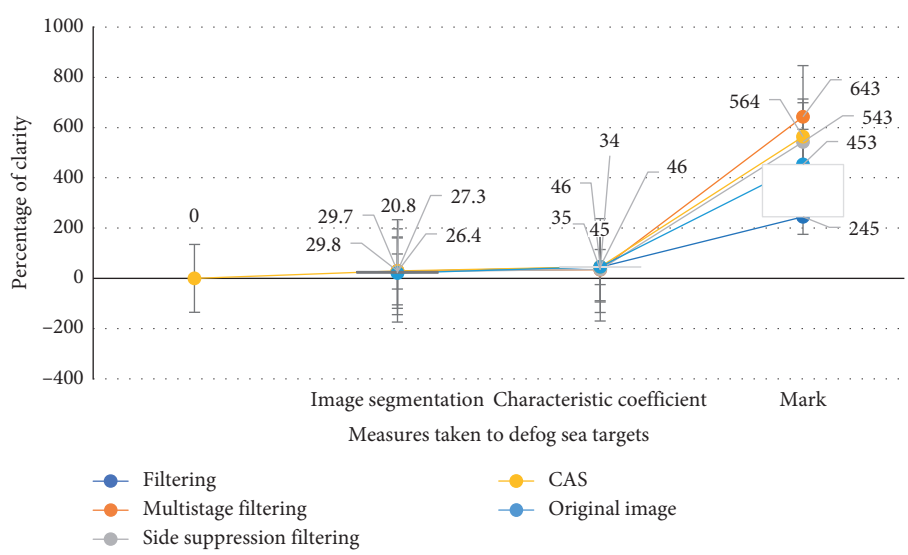


Figure 7: Percentage of sea fog image clarity under the algorithm.

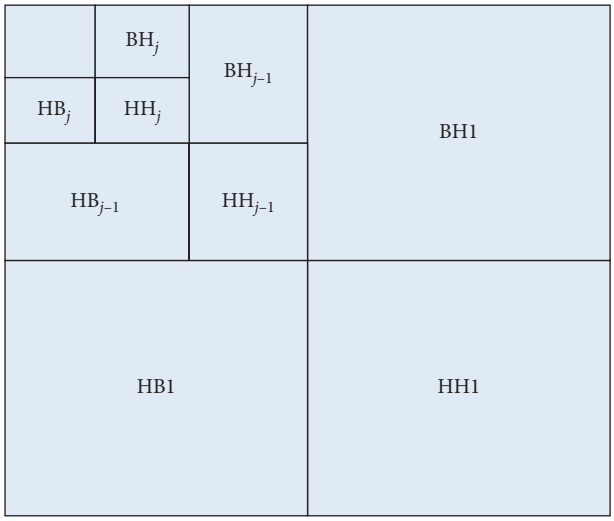


Figure 8: Speckle reduction of wavelet transform to suppress high-frequency subgraphs in different directions.

Table 3: Analysis of images collected at different time periods in a sea area (%).

	17–19 time period	19–21 time period	21–23 time period	23-1 time period	1–3 time period
Dark primary colors measure the clarity of the image	56	78	65	76	78
Percentage of atmospheric light value	35	47	58	56	39
Percentage of image clarity of atmospheric scattering model	47	46	78	56	89

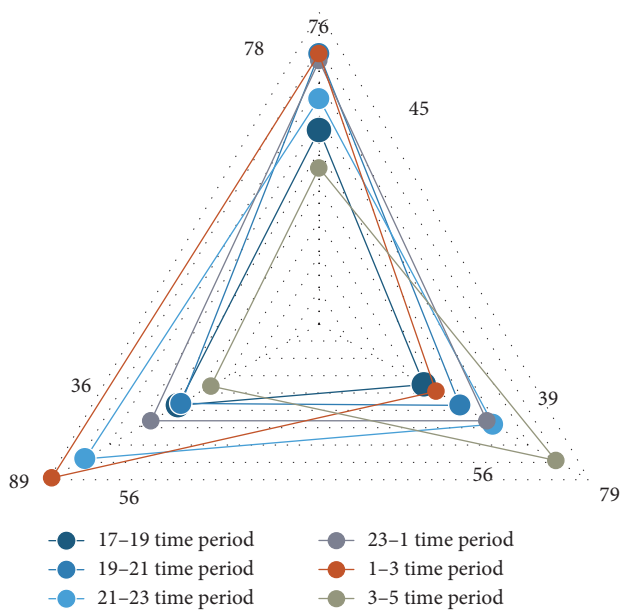
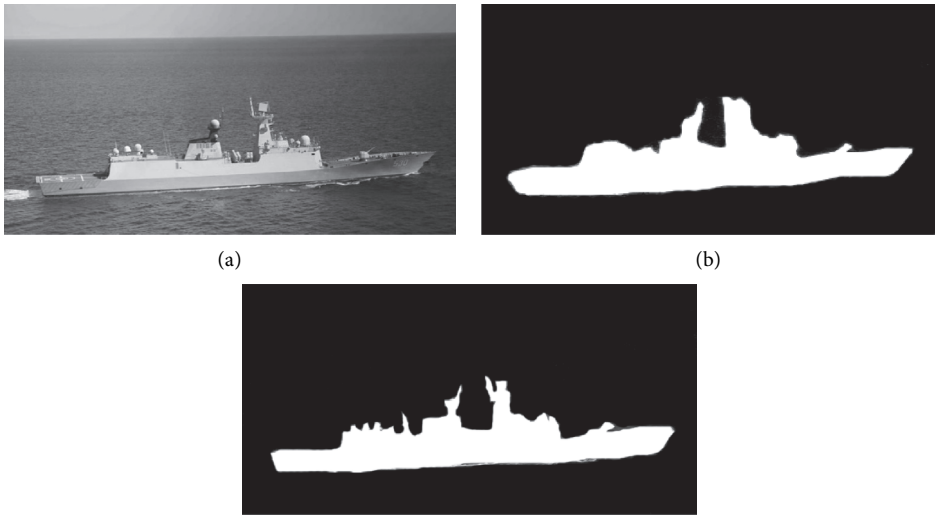


Figure 9: Analysis of images collected at different time periods in a sea area (%).

by the fog. According to the humidity and temperature of the air in the weather of the day, the percentage of the atmospheric light value is estimated, and the two are brought back to the atmospheric scattering model to obtain the image clarity after defogging percentage. Since the target single-frame detection method can effectively suppress background clutter interference in the conventional sea environment, while only a small amount of background clutter remains, it can achieve accurate detection of different sea targets with a local contrast of not less than 0.07. It can be seen from the figure that the overall clarity between 23 o'clock and 1 o'clock in the morning is high, and the clarity is



(a) (b) (c)

Figure 10: The step effect diagram combined with visual significance segmentation. (a) Original image. (b) Adaptive segmentation. (c) Grabcut image segmentation.

about twice the percentage of other time periods. The degree of clarity is higher than other times, so when using the atmospheric scattering model, a clear comparison is made.

4.3. Effect Comparison of Segmentation. In order to solve this problem, the prior information in the calculation of saliency map is fused into the energy function to improve the accuracy of segmentation. The improved calculation of energy function is shown as follows:

$$E_2(x) = \sum_{i \in v} \Phi(x_i)\varphi(x_i) + \beta \sum_{i,j \in N} \phi(x_i, x_j). \tag{11}$$

Among them, $\Phi(x_i)$ is the significant value after normalization, $\varphi(x_i)$ is the energy equation of regional term, and $\phi(x_i, x_j)$ is the energy of boundary constraint term. In order to solve the problem of the integrity of segmentation, after each iteration, the objects from the previous segmentation are expanded and corroded to get a new ternary image for the next iteration. Figure 10 is the step effect diagram combined with visual significance segmentation, in which Figure 10(b) is the result of adaptive segmentation and Figure 10(c) is the Grabcut segmentation.

5. Conclusions

At present, the Grabcut algorithm requires manual intervention and uses the method of manually labeling the rectangular frame to obtain the approximate area of the target. When processing complex images, the user cannot effectively label the rectangular frame. Based on the existing network model, this paper proposes a deep learning sea surface target detection and segmentation algorithm, and this method mainly uses the deep learning sea target detection and target detection model provided by Google and combines the superpixel segmentation algorithm to optimize the Grabcut algorithm, which realizes the deep learning sea target detection and segmentation algorithm, which can be accurate target contour and semantic information.

In the search and rescue of distressed targets in the deep sea, the ability to quickly and accurately find the distressed target is the determinant of the success of the search and rescue. Therefore, the detection and tracking technology of infrared targets has always been a hot spot for domestic and foreign scholars. Research on deep learning sea target detection and segmentation algorithms found that location prediction and category prediction in target detection tasks have different requirements for feature map features. Location prediction requires feature maps to contain rich details and positioning information, while category prediction requires feature maps to have abstraction semantic features and translation invariance. However, the existing algorithms use the same feature map for prediction in both position prediction and category prediction. To solve this problem, this paper redesigned the network structure based on the idea of separating the corresponding feature maps of location prediction and category prediction and the deep learning sea target detection and segmentation algorithm model. Two channels are added to the deep learning sea target detection and segmentation algorithm model to predict the location and category, respectively, to achieve a certain degree of separation of the two tasks. The feature map used for position prediction is merged with the low-level feature map through downsampling, which adds rich details and positioning information.

Through the above research work, the contrast of the infrared image of the sea target scene is significantly improved, and the details are clearer. In this paper, the wavelet transform-based filtering method is used for speckle noise suppression, and the deep learning-based method is used for land masking. The target detection part uses an improved

CFAR cascade algorithm and finally selects the best separable features for false alarm removal. In order to further illustrate the feasibility of the scheme, this paper uses measured data and simulation data to verify the scheme and discusses the effect of different signal-to-noise ratio, sea target type, and attitude on the algorithm performance. The research data shows that the deep learning sea target detection and segmentation algorithm has good detection performance and is generally applicable to ship target detection of different types and different attitudes. The results show that the deep learning sea target detection and segmentation algorithm fully takes into account the irregular shape and texture of the interfering target detected in the optical remote sensing image so that the accuracy rate is 32.7% higher and the efficiency is increased by about 1.3 times. The deep learning sea target detection compared with segmentation algorithm has strong target characterization ability and can be applied to ship targets of different scales.

Data Availability

All the data in the article are real and available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Hainan Provincial Natural Science Foundation of China (519QN180), Hainan Provincial Key R&D Plan (ZDYF2019014), National Natural Science Foundation of China and Macau Science and Technology Development Joint Fund (0066/2019/AFJ), and the program and the Scientific Research Foundation of Hainan University (KYQD(ZR)1859).

References

[1] G. Gao, K. Ouyang, Y. Luo, S. Liang, and S. Zhou, "Scheme of parameter estimation for generalized gamma distribution and its application to ship detection in SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 3, pp. 1812–1832, 2017.

[2] D. Wang, F. X. Cui, and Z. Z. ShiJiang, "Multi-feature sea-land segmentation based on pixel-wise learning for optical remote-sensing imagery," *International Journal of Remote Sensing*, vol. 38, no. 15, pp. 4327–4347, 2017.

[3] Y. He and L. Yao, "Space-based sea target information awareness and fusion," *Geomatics & Information Science of Wuhan University*, vol. 42, no. 11, pp. 1530–1536, 2017.

[4] A.C. Wölfl, J. Jencks, and G. Johnston, "Identifying deep-sea target areas for a pilot Atlantic seabed mapping project using GIS techniques," *Journal of Ocean Technology*, vol. 12, no. 4, pp. 28–42, 2017.

[5] X. Qiao, L. J. Bao, and D. J. LiZou, "An automatic active contour method for sea cucumber segmentation in natural underwater environments," *Computers and Electronics in Agriculture*, vol. 135, pp. 134–142, 2017.

[6] R. Fu, B. Li, Y. Gao, and P. Wang, "Fully automatic figure-ground segmentation algorithm based on deep convolutional neural network and grabcut," *Iet Image Processing*, vol. 10, no. 12, pp. 937–942, 2016.

[7] R. Liu, D. Wang, and P. Jia, "Overview on small target detection technology in infrared image," *Laser & Optoelectronics Progress*, vol. 53, no. 5, Article ID 050004, 2016.

[8] Y. Ren, "An image segmentation algorithm based on K-means algorithm and otsu," *Boletin Tecnico/technical Bulletin*, vol. 55, no. 9, pp. 78–83, 2017.

[9] O. Paula López, F. Laura Docío, and M. Carmen García, "SEA_AP: a segmentation and labelling tool for prosodic analysiS," *Dialectologia*, no. 6, pp. 223–244, 2016.

[10] K. Yan, Y. Bai, and H. C. Wu, "Robust target detection within sea clutter based on graphs," *IEEE Transactions on Geoence and Remote Sensing*, vol. 57, no. 99, pp. 1–11, 2019.

[11] L. Sun, Y. Li, and Y. Zou, "Pig image segmentation method based on improved graph cut algorithm," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 33, no. 16, pp. 196–202, 2017.

[12] Y. Yanfeng, Y. Wang, P. Liu, H. Luo, B. Cheng, and H. Sun, "Infrared LSS-target detection via adaptive TCAIE-LGM smoothing and pixel-based background subtraction," *Photon Sensor: English Version*, vol. 9, no. 2, pp. 179–188, 2019.

[13] L. Lu and J. Wang, "Image processing and recognition algorithm for target tracking," *International Journal on Smart Sensing and Intelligent Systems*, vol. 9, no. 1, pp. 353–376, 2016.

[14] W. Yu, Y. Wang, and H. Liu, "Superpixel-based CFAR target detection for high-resolution SAR images," *IEEE Geoence and Remote Sensing Letters*, vol. 13, no. 5, pp. 1–5, 2016.

[15] J. Enyuan and W. Xuejun, "Vehicle abnormal behavior detection based on trajectory tracking," *Journal of Jilin University (Information Science Edition)*, vol. 34, no. 1, pp. 98–103, 2016.

[16] A. N. Katulev and A. A. Khramichev, "Automatic wavelet-based segmentation of a background-and-target frame from an optoelectronic device for detection of dynamic objects in 2D images," *Journal of Optical Technology*, vol. 83, no. 2, p. 98, 2016.

[17] L. P. Dorado-Munoz and D. W. Messinger, "Initial study of schroedinger eigenmaps for spectral target detection," *Optical Engineering*, vol. 55, no. 8, Article ID 083101, 2016.

[18] Z. Yuan, Z. Jianming, and L. Xiao, "Infrared small target detection using weighting LoG operator," *Journal of Applied Optics*, vol. 38, no. 1, pp. 114–119, 2017.

[19] P. Lv, C. S. Sun, and G. Liu, "A method for weak target detection based on human visual contrast mechanism," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 2, pp. 261–265, 2019.

[20] K. Liu, "Research on improved fingerprint image compression and texture region segmentation algorithm," *International Journal Bioautomotion*, vol. 21, no. 3, pp. 231–240, 2017.

[21] Q. H. Wu, J. Wu, and L. Zhu, "Image segmentation algorithm based on graph theory and FCM," *Chinese Journal of Liquid Crystals & Displays*, vol. 31, no. 1, pp. 112–116, 2016.

[22] Y. Liu and S. Hu, "Blind restoration based on detection and segmentation of motion blurred image," *Xi Tong Gong Cheng Yu Dian Zi Ji Shu/Systems Engineering and Electronics*, vol. 39, no. 3, pp. 662–667, 2017.

[23] B. Gao, X. Li, W. L. Woo, and G. Y. Tian, "Physics-based image segmentation using first order statistical properties and genetic algorithm for inductive thermography imaging," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2160–2175, 2018.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1820527, 12 pages
https://doi.org/10.1155/2020/1820527

Research Article

Cooperative Multiagent Deep Deterministic Policy Gradient (CoMADDPG) for Intelligent Connected Transportation with Unsignalized Intersection

Tianhao Wu,[1] Mingzhi Jiang,[1] and Lin Zhang [2,1]

[1]*School of Information & Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Beijing Information Science and Technology University, Beijing 100192, China*

Correspondence should be addressed to Lin Zhang; zhl@bistu.edu.cn

Received 20 April 2020; Revised 9 June 2020; Accepted 23 June 2020; Published 22 July 2020

Guest Editor: Chi-Hua Chen

Unsignalized intersection control is one of the most critical issues in intelligent transportation systems, which requires connected and automated vehicles to support more frequent information interaction and on-board computing. It is very promising to introduce reinforcement learning in the unsignalized intersection control. However, the existing multiagent reinforcement learning algorithms, such as multiagent deep deterministic policy gradient (MADDPG), hardly handle a dynamic number of vehicles, which cannot meet the need of the real road condition. Thus, this paper proposes a Cooperative MADDPG (CoMADDPG) for connected vehicles at unsignalized intersection to solve this problem. Firstly, the scenario of multiple vehicles passing through an unsignalized intersection is formulated as a multiagent reinforcement learning (RL) problem. Secondly, MADDPG is redefined to adapt to the dynamic quantity agents, where each vehicle selects reference vehicles to construct a partial stationary environment, which is necessary for RL. Thirdly, this paper incorporates a novel vehicle selection method, which projects the reference vehicles on a virtual lane and selects the largest impact vehicles to construct the environment. At last, an intersection simulation platform is developed to evaluate the proposed method. According to the simulation result, CoMADDPG can reduce average travel time by 39.28% compared with the other optimization-based methods, which indicates that CoMADDPG has an excellent prospect in dealing with the scenario of unsignalized intersection control.

1. Introduction

In recent years, the development of connected vehicles [1] has prompted the innovation of transportation technology. By introducing the technologies of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, multivehicle coordination has become possible to improve traffic safety and efficiency. This makes the cooperative intelligent transportation systems (C-ITS) a new research hotspot [2].

As a classic application scenario of Intelligent Connected Transportation, the intersection is more complicated and challenging for the cooperation of intervehicle control than the highway. In the intersection, vehicles will enter from different directions and leave through different exits. The conflict between different vehicles will greatly limit the efficiency and safety of the intersection traffic. Therefore, a flexible and complicated intersection control system is necessary. There have been several related works focusing on traffic signal timing optimization. Ge et al. proposed a cooperative method for multi-intersection signal control based on Q-learning with Q-value transfer [3]. Kim and Jeong proposed a cooperative traffic signal control scheme using traffic flow prediction for a multi-intersection [4]. Kamal et al. concentrated on the mixed manual-automated traffic scenario and developed an adaptive traffic signal control scheme for an isolated intersection [5]. Yu proposed a fuzzy programming-based approach to optimize the signal timing for an isolated intersection, which used operation efficiency, traffic capacity, and exhaust emission as the joint optimization goals [6]. Bai et al. proposed a scenario where trams

crossed intersections without stop and presented a coordinated control model along the tramline to optimize the control of prioritized traffic signals [7]. Xu et al. proposed a game-based policy for the signal control of an isolated intersection together with rerouting of the vehicles using vehicle-to-infrastructure communication [8]. Lu et al. presented a novel speed control method for the successive signalized intersections under connected vehicles environment. With the timing information and vehicle queues at the signalized intersection, vehicle speed was optimized to reduce fuel consumption and emissions [9]. Xu proposed a cooperative method to optimize roadside signal optimization and on-board vehicle speed control at the same time [10]. In addition, with the development of autonomous driving, the mixed traffic of manual-automated driving coexistence is also attracting researchers' attention. Boudaakat et al. established a hydrodynamic model to calculate the total capacity of roundabouts and provided a method to control traffic congestion [11]. Kamal et al. presented a novel adaptive traffic signal control scheme, aiming at minimizing the total crossing time of all vehicles and ensuring comfortable crossing of manually driven vehicles [12]. Gupta et al. proposed a conceptual model for negotiation between self-driving vehicles and pedestrians, which shows an improvement in the overall travel time of the vehicles as compared with the current best practice behaviour, always stop, of autonomous vehicles [13]. In general, traffic signal control schemes should maximize traffic efficiency under the premise of ensuring traffic safety. To make the intersection traffic control more accurate and in real time, many researchers have begun to focus on the study of unsignalized intersection control.

Generally, the researchers divide unsignalized intersection control schemes into two categories, centralized and distributed. Centralized coordination approaches collect the global information of the entire intersection to regulate the vehicles at the intersection. Dai et al. solve the intersection control problem with convex optimization [14]. Guan et al. proposed a centralized conflict-free cooperation method for multiple connected vehicles at unsignalized intersection using model accelerated proximal policy optimization [15]. Qian et al. devised the AIDM algorithm to schedule a vehicle or a platoon to pass the unsignalized intersection [16]. Nevertheless, these centralized schemes are pressed for communication and computation because all vehicles need a central controller to dispatch them.

In decentralized coordination approaches, the centralized controller disappears, and the controller is configured separately in each vehicle to adjust its trajectories considering kinetic information and conflict relationship of adjacent vehicles. Xu et al. introduced a conflict-free geometry topology and designed a distributed controller to stabilize the involved vehicle at the intersection [17]. Bian et al. divided the intersection area into four areas according to the distance and present a process containing observation, optimization, and control [18]. Hadjigeorgiou and Timotheou studied the optimization of the travel time and fuel consumption balance of autonomous vehicles through unsignalized intersection [19]. Belkhouche proposed decentralized multiagent control laws, which are derived for conflict resolution between vehicles [20]. Hsu et al. studied the interaction between vehicles and pedestrians at unsignalized intersections and proposed a decision theory model to represent the interaction [21]. Wang et al. proposed a cooperative algorithm to transform the high-dimensional problem of cooperative driving for multiple vehicles at multiconflict points into the single-dimensional problem of searching the optimal time for vehicles to enter the intersection [22]. Yang and Oguchi developed an advanced vehicle control system with connected vehicles to reduce vehicles delays under a partially connected environment [23]. However, well-designed models and controllers can only show interpretable parts, and more hidden parts become bottlenecks in performance improvement.

One of the most critical goals in artificial intelligence is to obtain a new skill, especially in a multiagent environment. Reinforcement learning (RL) can improve the policy via trial-and-error interaction with the environment, which is analogous to human beings. Recently, RL has taken an essential role in a variety of fields, for example, wireless communication [24] and autonomous driving [25]. Mnih et al. proposed Deep Q-learning Network (DQN) and obtained superhuman performance on Atari video games [26]. Considering that DQN is only applicable to the problem with discrete action spaces, Deep Deterministic Policy Gradient (DDPG) is proposed to solve continuous control problems [27]. When the scenario extends from a single agent to multiple agents, more information can be taken into consideration to improve algorithm performance. Multiagent DDPG (MADDPG) is a multiagent policy gradient algorithm where agents learn a centralized critic based on the observation and actions of all agents [28]. There have already been many applications in the field of intersection control. Liang et al. proposed a double-dueling deep Q-network to control the traffic light cycle [29]. Zhou et al. proposed a car-following model, based on reinforcement learning, to obtain an appropriate driving behaviour to improve travel efficiency, fuel consumption, and safety at signalized intersections [30]. Lee et al. employed reinforcement learning that recognizes an entire traffic state and jointly controls all the traffic signals of multiple intersections [31]. However, the uncertainty of agent number poses a further challenge to address some problems, such as distributed coordination at unsignalized intersection.

This paper proposes a distributed conflict-free cooperation method, Cooperative MADDPG (CoMADDPG), for multiple connected vehicles at unsignalized intersection. The main contributions of this paper are as follows:

(i) This paper formulates the scenario of multiple vehicles passing through an unsignalized intersection as a multiagent reinforcement learning problem.

(ii) The Cooperative MADDPG (CoMADDPG) is proposed, which modifies the classic MADDPG algorithm to adapt the dynamic quantity agents. In CoMADDPG, each vehicle selects reference vehicles to construct a partial stationary environment, which is necessary for the introduction of the RL method.

(iii) This paper also proposes a novel vehicle selection method, which can project all reference vehicles on a virtual lane and selects the largest impact vehicles on the virtual lane. It can assist the CoMADDPG algorithm to converge quickly and avoid collisions effectively.

The rest of this paper is organized as follows. Section 2 illustrates our problem statement and presents the settings of states, actions, and rewards. Section 3 introduces the preliminaries of multiagent reinforcement learning and the workflow of the proposed CoMADDPG algorithm. Section 4 presents the experimental settings and results. Section 5 concludes this work.

2. Problem Statement and Formulation

2.1. Problem Statement. This paper focuses on a 4-direction intersection shown in Figure 1. Each direction denotes the location in the figure, that is, up, down, left, and right, respectively. A certain distance of the intersection is focused on. There are 4 entrances and 4 exits in total, which are unsignalized, and each direction contains only one lane. The vehicles are only allowed to go straight.

As depicted in Figure 1, boxes in different colours represent vehicles in different lanes. The trajectories of the vehicles intersect into 4 conflict points in the merging zone of the intersection. Based on the collected information, each vehicle independently decides its acceleration and deceleration using a policy network. When the vehicles enter the intersection area, the centralized server distributes the newest model to them. During the running process of the vehicles, they produce so-called experience, recording the running data of themselves and reference vehicles at each time step. The reference vehicles are selected via a proposed vehicle selection method. The whole process is continuous, with vehicles entering and leaving the intersection area.

Several assumptions are adopted as follows. Firstly, the kinetic information of vehicles can be measured to support the decision made by each vehicle. Then, it is assumed that all approaching vehicles are connected and automated so that each vehicle can strictly obey the planned acceleration, adjust the velocity, and pass the intersection automatically. Additionally, all vehicles enter the intersection according to the set time, which corresponds to the Poisson process.

2.2. MARL Formulation. The problem is formulated as a multiagent reinforcement learning problem, and each vehicle is treated as an agent by defining state space, action space, and reward function.

2.2.1. State and Action Space. According to the assumptions, each vehicle can obtain others' kinetic information via vehicle-to-vehicle communication. To achieve the aim of collaboration, the state of a vehicle needs to include the dynamics of its adjacent vehicles. However, the dimensions of the state cannot be arbitrarily expanded, so several vehicles with the largest impact are chosen, as is shown in equation (1). The subscript m indicates the maximum number of the largest impact vehicles under consideration. As for the definition of $s^i_{\text{other},j}$, the subscript j represents the jth largest impact vehicle of vehicle i, and it has no relationship with the identity of any vehicle:

$$s_i = \left\{s^i_{\text{own}}, s^i_{\text{other},1}, \ldots, s^i_{\text{other},j}, \ldots, s^i_{\text{other},m}\right\}, \tag{1}$$

$$s^i_{\text{own}} = \left\{p^i, v^i, a^i\right\}, \tag{2}$$

$$s^i_{other,*} = \left\{p^i_*, v^i_*, a^i_*\right\}. \tag{3}$$

In equation (2), s^i_{own} represents the state of vehicle i, including position, velocity, and acceleration. In equation (3), $s^i_{other,*}$ represents the reference vehicles of vehicle i, including position, velocity, and acceleration. Commonly, the position can be formed with Cartesian coordinate, that is, (x, y). However, through the analysis of the task formulation, conflicting vehicles at the same distance from the intersection will have a high correlation. Therefore, polar coordinate (μ, θ) is utilized instead of (x, y). There are only 4 directions in the problem, so the position is denoted by (d, l). Herein, d is the distance from the vehicle to the conflict point, and it is positive when approaching the intersection and negative when leaving. l is the index of lane, that is, $\{1, 2, 3, 4\}$.

2.2.2. Reward Settings. The reward function is designed, as shown in Table 1. Compared with reward settings in [15], the reward is not defined with the final situation, such as vehicle passing or collision but scattered in the running process. Here, the distance difference Diff_D and the time difference Diff_T are used to assist each vehicle to obtain its reward:

$$\text{Diff}_D = \left|p_{\text{cur}} - p_{\text{ref}}\right|, \tag{4}$$

$$\text{Dif}f_T = \frac{p_{\text{cur}} - p_{\text{ref}}}{v_{\text{cur}} - v_{\text{ref}}}. \tag{5}$$

In equations (4) and (5), p_{cur} and p_{ref} represent the positions of the current vehicle and reference vehicle. v_{cur} and v_{ref} are the velocities of the current vehicle and reference vehicle. We set the end of the entering lane to zero. If Diff_T is greater than zero, the current vehicle and reference vehicle are approaching, and vice versa.

It is noted that as the distance difference shrinks, the risk factor does not change linearly, and a logarithmic function is introduced to describe this nonlinear change. On the other hand, the sign of time difference can be a good indicator of whether the distance between the two vehicles is increasing or decreasing. If the distance between vehicles is small enough, the increasing distance will generate positive rewards and vice versa. The transformation of the hyperbolic tangent function is a good description of this change. Moreover, to avoid reward expansion, which is an important issue in RL, we limit the reward value after each calculation and reward is controlled at $[-20, 20]$.

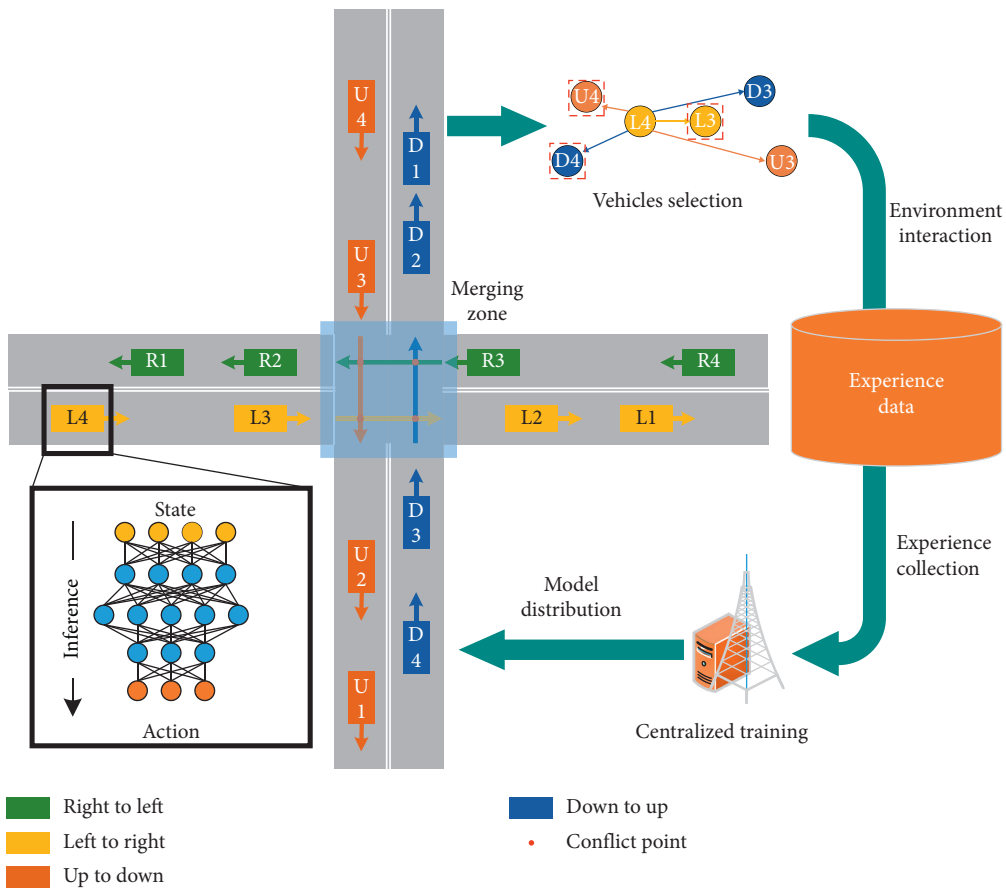


Figure 1: The overall flow diagram of cooperative reinforcement learning integrated with intersection scenario.

Table 1: Reward settings.

Reward items	Reward
Distance difference	$\log(\mathrm{Diff}_D)$
Time difference	$1/\tanh(-\mathrm{Diff}_T)$

3. Cooperative Multiagent Deep Deterministic Policy Gradient

In this paper, partially observable Markov games are considered, constituting a multiagent Markov decision process. The possible state $\mathcal{S}$, a set of actions $\mathcal{A}_1, \ldots, \mathcal{A}_N$, and a set of observations $\mathcal{O}_1, \ldots, \mathcal{O}_N$ jointly describe a Markov game for N agents. To determine the action, each agent utilizes a stochastic policy $\pi_{\theta_i}: \mathcal{O}_i \times \mathcal{A}_i$, which outputs the next state based on the state transition function $\mathcal{T}: \mathcal{S} \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_N \mapsto \mathcal{S}$. After interacting with the environment, each agent obtains reward with the function of state and action $\mathcal{S} \times \mathcal{A}_1 \mapsto \mathbb{R}$ and gets a separate observation o_i. The initial state is dependent on a distribution ρ. Each agent runs for maximizing its expected return $R_i = \sum_{t=0}^{T} \gamma^t r_i^t$, where T is the time horizon and γ is a discount factor.

3.1. Multiagent Deep Deterministic Policy Gradient. A significant problem faced by the traditional RL algorithm is that each agent is learning to improve the policy continuously. Thus, from the perspective of each agent, the environment is dynamic, which is not stationary for traditional RL algorithm. To a certain extent, it is impossible to adapt to the dynamic environment by merely changing the agent's policy. Due to the instability of the environment, the critical techniques of DQN-like experience replay cannot be directly used. The policy gradient exacerbates the problem of significant variance due to the increase in the number of agents.

Then MADDPG is an adaptation of actor-critic methods which considers action policies of other agents and can learn policies that require complex multiagent coordination. The algorithm has three characteristics. Firstly, the optimal policy obtained through learning can produce optimal action using only local information. Secondly, there is no need to know the requirements of the dynamic model of the environment and interagent communication. Thirdly, the algorithm can be used in both cooperative and competitive environments.

In the game with N agents, the policies on each agent can be represented as $\mu = \{\mu_1, \ldots, \mu_N\}$, and the gradient of the expected return for agent i, $J(\theta_i) = \mathcal{E}[R_i]$, can be written as

$$\nabla_{\theta_i} J(\boldsymbol{\mu}_i) = E_{x,a\sim\mathcal{D}}\left[\nabla_{\theta_i}\boldsymbol{\mu}_i(a_i \mid o_i)\nabla_{a_i}\mathrm{Q}_i^{\boldsymbol{\mu}}(\mathbf{x}, a_1, \ldots, a_N)\big|_{a_i=\mu_i(o_i)}\right], \tag{6}$$

where $\mathcal{D}$ is the experience replay buffer, which contains the tuples (o_i, o_i', a_i, r_i) to record the experience of each agent i.

$Q_i^{\mu}(\mathbf{x}, a_1, \ldots, a_N)$ denotes a centralized action-value function that takes the action of all agents and some state information $\mathbf{x}$ and outputs the Q-value for agent i. In some fundamental cases, $\mathbf{x}$ includes the observation of all agents: $\mathbf{x} = (o_1, \ldots, o_N)$. The centralized action-value function is updated as

$$y^i = r_i^j + \gamma Q_i^{\boldsymbol{\mu}'}\left(x'^j, a_1', \ldots, a_N'\right)\Big|_{a'_k = \mu'_k\left(o_k^j\right)}, \tag{7}$$

$$\mathscr{L}(\theta_i) = \mathscr{E}_{\mathbf{x},a,r,\mathbf{x}_l}\left(y^j - Q_i^{\boldsymbol{\mu}}\left(x^j, a_1^j, \ldots, a_N^j\right)\right)^2, \tag{8}$$

where $\boldsymbol{\mu}' = \left\{\mu_{\theta_1'}, \ldots, \mu_{\theta_i'}, \ldots, \mu_{\theta_N'}\right\}$ is the target policy set with delayed parameters θ_i'. The main intention of MADDPG is that when the actions produced by all agents are known, the environment is stationary even if the policies vary. Additionally, the algorithm has three techniques. Firstly, actor and critic constitute centralized training, and actor can run only by knowing local information during inference. Secondly, experience replay is improved to apply to a dynamic environment. Thirdly, policy ensemble is utilized to enhance stability and robustness.

3.2. Cooperative MADDPG (CoMADDPG). One of the challenges for multiagent reinforcement learning is when policies of agents are updated, the environment changes, which contradicts existing assumptions of a stationary environment. Accordingly, a primary motivation behind MADDPG is that the actions taken by all agents are known, which makes the environment be considered stationary even as the policies change. However, by default, different observation variables in MADDPG will correspond to the agents one by one, which significantly limits the application scenarios of this algorithm. In this paper, the definition of a stationary environment is extended to suit the situation of more agents.

In the problem of distributed vehicle control at the intersection, the fluent entry and exit of vehicles lead to an uncertain and large number of agents. From the perspective of a stationary environment, it can exist not only globally but also partially. To construct the environment, an agent selects several agents as reference agents. Therefore, the gradient of the expected return can be rewritten as

$$\nabla_{\theta_i} J(\boldsymbol{\mu}_i) = E_{x,a\sim\mathscr{D}}\left[\nabla_{\theta_i}\boldsymbol{\mu}_i\left(a_i \mid \vec{\mathbf{o}_\mathbf{i}}\right)\nabla_{a_i} Q_i^{\boldsymbol{\mu}}\left|\left(\vec{\mathbf{o}_\mathbf{i}}, \vec{\mathbf{a}_\mathbf{i}}\right)\right|_{a_i=\mu_i\left(\vec{\mathbf{o}_\mathbf{i}}\right)}\right]. \tag{9}$$

In equation (9), $\vec{\mathbf{o}_\mathbf{i}} = \left\{o_i, o_{i_1}, \ldots, o_{i_j}, \ldots, o_{i_m}\right\}$ and $\vec{\mathbf{a}_\mathbf{i}} = \left\{a_i, a_{i_1}, \ldots, a_{i_j}, \ldots, a_{i_m}\right\}$ denote the set of observations and the set of actions, which come from agent i and reference agents $\{i_1, \ldots, i_j, \ldots, i_m\}$. Note that the entity of each reference agent i_j may change at each time step. For instance, there are four vehicles $\{A, B, C, D\}$ moving in the same lane in sequence. If vehicle A is the current vehicle, the set of observations $\vec{\mathbf{o}_\mathbf{A}}$ can be written as $\{o_A, o_B, o_C, o_D\}$. When vehicle B overtakes vehicle C, the sequence of the vehicles becomes {A, C, B, D}, and the set of observations $\vec{\mathbf{o}_\mathbf{A}}$ becomes $\{o_A, o_C, o_B, o_D\}$. When this data enters the neural network as input, the position of the variables will play an important role. The above operation can decouple the identity and running information of the vehicles so that the proposed CoMADDPG can be adapted to the scenario of more agents.

Note that more agents will not make the decision process more complex. This is because although more agents will expand the length of the input data, those data only need to pass through the same network structure. Moreover, we introduce the virtual lane to propose a vehicle selection method, which makes the current vehicle only care about the largest impact vehicles. In order to ensure the effectiveness of small-scale neural networks, we set an upper limit on the number of reference vehicles. All agents play the same role in cooperation with each other. In the stage of decision-making, the running states of the current vehicle and reference vehicles are combined into a set. Each action takes into account the states and actions of the reference vehicles. The vehicle selection method is explained in detail in the next section.

3.3. Largest Impact Vehicles Selection. The proposed method is built based on a distributed system, and all vehicles are regarded as the equal agents in the system. The process of selecting the largest impact vehicles is a prestep for information gathering. Based on the decoupling of identity and running information of the vehicles, how to select vehicles to obtain running information becomes an issue. There are two main types of vehicle collisions near intersections. One is the longitudinal collision in the lane, and the other is the lateral collision at the merging zone. The longitudinal collisions can be resolved by selecting adjacent vehicles in the same lane. To solve the lateral collision, this paper introduces the concept of the virtual lane.

For clear description, eight vehicles (L3, L4, R3, R4, U3, U4, D3, and D4) are chosen for illustration in Figure 2. At each time step, each vehicle will perform the vehicles selection. Here, vehicle L4 is taken as an example, so the lane from left to right is considered as the baseline of the virtual lane. Then vehicles U3, U4, D3, and D4 are projected onto the virtual lane. Because the up-to-down lane and the down-to-up lane have different conflict point with the left-to-right lane, the vehicles would have different offsets when projected. Since the right-to-left lane does not conflict with the left-to-right lane, R3 and R4 do not need to be projected. In Figure 2, the radius of the arc represents the distance to the conflict points. The dots with different colours represent the projected vehicle on the virtual lane, which is a black arrow pointing forward. In the scenario of this paper, there are four entering lanes, so four virtual lanes take effect. The projected virtual platoon is shown in Figure 3, and different colours and arrows indicate different direction.

In this paper, the selection of the largest impact vehicles can depend on space distance. When L4 is performing the vehicle selection, a star structure is obtained to express the relationship between L4 and other vehicles in Figure 4. Moreover, the length of the link between L4 and other dots means the relationship among vehicles. The shorter the

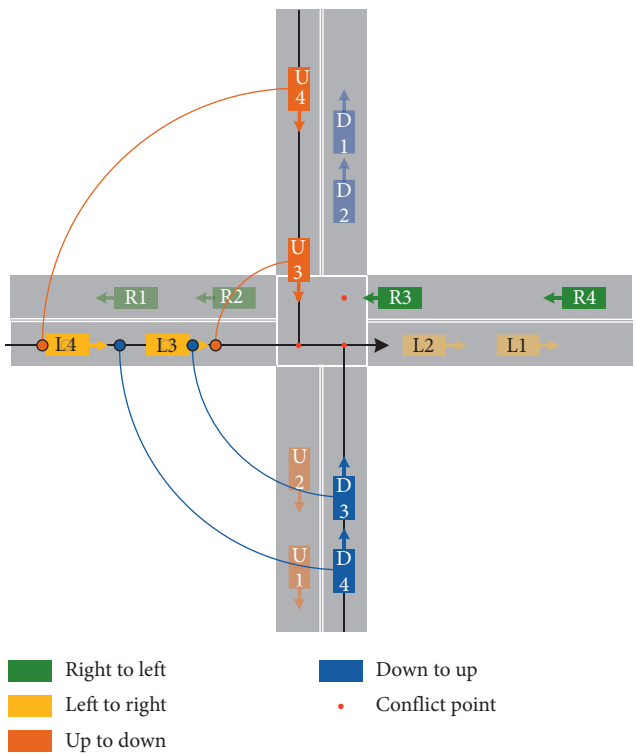


Figure 2: Virtual platoon projection. L3, L4, R3, R4, U3, U4, D3, and D4 are chosen for illustration, and L4 is taken as an example to perform the vehicles selection.

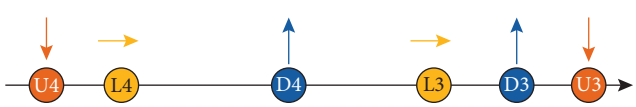


Figure 3: A projected virtual platoon.

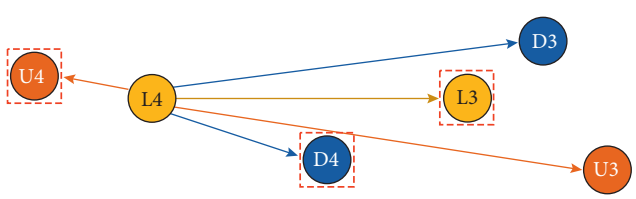


Figure 4: Star structure for L4. The shown dots represent candidate vehicles, and the red-dashed boxes select the vehicles to be considered.

length of the directed line, the larger the impact. Finally, among the linked dots to L4, several largest impact vehicles are selected with red-dashed boxes. The states, including positions and velocities, of these selected vehicles would be taken into the process of model training and inference.

With the largest impact vehicle selection, the proposed CoMADDPG can use a relatively simple network structure to handle a large number of agents.

3.4. Algorithm Architecture. This part illustrates how to apply CoMADDPG algorithm to this distributed control problem at intersection.

A learning algorithm for this distributed RL problem consists of two main parts: CoMADDPG trainer and executor. Figure 5 shows the overall architecture. The executor is applied to get updated policy from the trainer and uses it to collect experience from the simulation environment. Then, the trainer uses experience data from the executor to update policy network, which follows the DDGP model update process. Finally, the trainer distributes the newest model to the executors.

4. Experiments

4.1. Experimental Settings. In this section, CoMADDPG is trained and evaluated in the scenario of the intersection, which contains 4 different directions and allows vehicles to go straight without turning. Therefore, there are four conflicting points in the intersection, and each conflict point corresponds to a virtual lane. Furthermore, there are 4 types of vehicles, and each type possesses the same entry and exit. Vehicles appear at the beginning of each entering lane and follow a Poisson process with different vehicle density. Here, with the predefined arrival time and initial velocity, there is no need to set the distance between vehicles. Geometry and vehicle dynamics parameters are listed in Table 2. As for velocity and initial velocity, m/s is used in the experiment, but, in order to facilitate understanding, km/h is used in this paper. The central server on the intersection can collect experience from the vehicle, update the model, and distribute the newest model to the vehicles entering the lane. After receiving the newest model, the vehicle could determine the action with the model and send the produced experience to the central server. For results, the training processes of CoMADDPG are shown and illustrate our improvement in MADDPG.

4.2. Implementation Details. In CoMADDPG, there are two modules, actor and critic, for inference and training. Each module corresponds to a network structure without shared parameters, which are shown in Figure 6. In Figure 6(a), the actor module inputs state and outputs action, which contains three dense layers and two normalization layers, and chooses ReLU, presented in equation (10), as activation function. On the other hand, the critic module is used to evaluate the actions with Q value in a specific state. The structural difference between critic and actor lies in action set, which is concatenated to the processed state as an additional input. In CoMADDPG, action set contains the actions performed by the current vehicles and reference vehicles:

$$\mathrm{ReLU}(x) = \begin{cases} x, & x > 0, \\ 0, & x \leq 0. \end{cases} \tag{10}$$

Furthermore, complete hyperparameters are listed in Table 3.

4.3. Results and Discussion. This section presents the performance of our algorithm at the intersection and analyzes

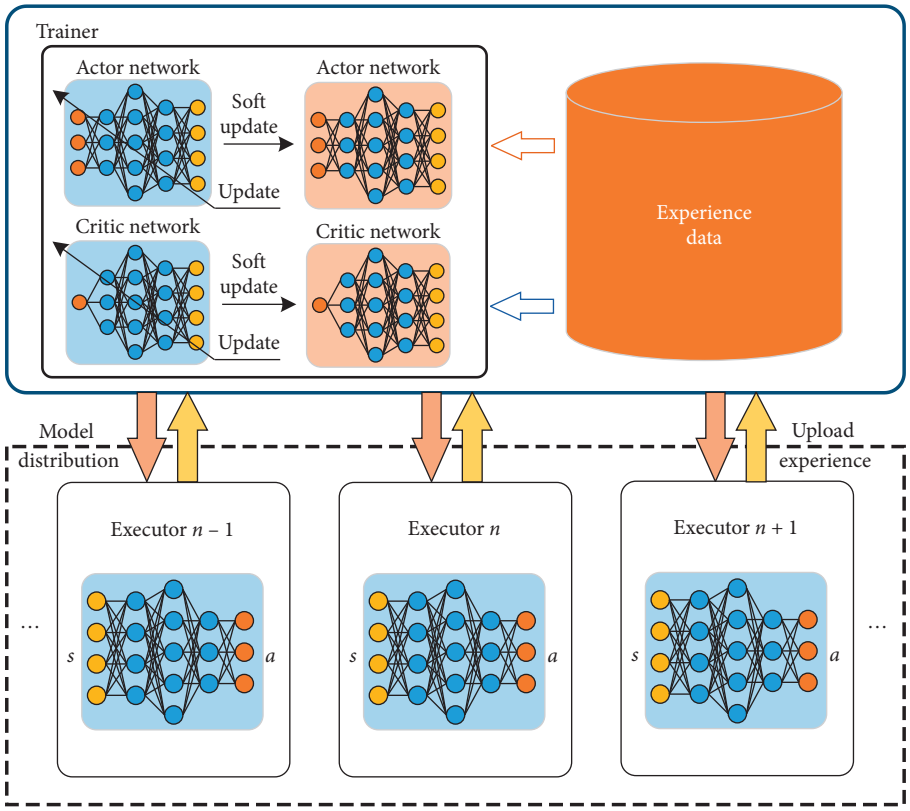


Figure 5: Overall architecture of the algorithm. The new executor, that is, vehicle, downloads the newest network from the centralized trainer. The centralized trainer updates the networks with the experience from executors.

Table 2: Geometry and vehicle dynamics parameters.

Parameter	Value
Lane length (m)	155
Vehicle size (m)	2
Velocity (km/h)	[18,50]
Initial velocity (km/h)	36
Acceleration (m/s^2)	[−3,3]

the empirical results. Firstly, a 3D view is used to visually show the changes in the position and velocity of vehicles as they pass through the intersection. Then, the process of training is observed to verify the convergence of the proposed CoMADDPG. Next, as one of the significant contributions, the virtual lane is compared with actual lanes under different parameters. Moreover, the average travel time is tested with different vehicle densities and different lane lengths. Finally, an optimization-based method is chosen for comparison, and the real-time performance is discussed.

In order to clearly understand the running state, Figure 7 illustrates the position and velocity profiles of approaching vehicles from 4 entrances of the intersection. In Figure 7(a), the approaching vehicles entered the merging zone of intersection orderly. Figure 7(b) presents approaching vehicles to adjust their velocities to achieve collision-free. The speed adjustment is too frequent, which will be optimized to improve the stability of the vehicle speed in our future work.

Firstly, to prove the ability of convergence, the experiment is designed to compare the training process between CoMADDPG and DDPG. From the perspective of reinforcement learning model training, the loss function, reward, and the statistics of the number of collisions are measured. CoMADDPG is the adaptation of MADDPG in the case of dynamic quantity agents, and DDPG is employed as our baseline, which only considers its actions without others' actions in the stage of centralized training. In terms of the mean reward and the loss function of the actor, there is no apparent difference between the two algorithms. In Figure 8(a), the loss functions of actor, which is the decision module in CoMADDPG and DDPG, cross down and tend to be flat, which means both algorithms can converge. During the most steps of the training, DDPG has lower loss than the proposed CoMADDPG, which is due to the simpler state and action input. However, the loss function is only an auxiliary indicator, and the number of collisions in Figure 8(d) is more valuable. In Figure 8(b), DDPG shows a higher loss than CoMADDPG. This is due to the insufficient information obtained by DDPG, which cannot form a stationary partial environment to perform reinforcement learning algorithm. In Figure 8(c), both algorithms have the same trend in the change of the mean reward. Concerning the cumulative number of collisions during training, CoMADDPG is much lower than DDPG in Figure 8(d). Because the CoMADDPG module can still explore during the training process, the cumulative collision curve of

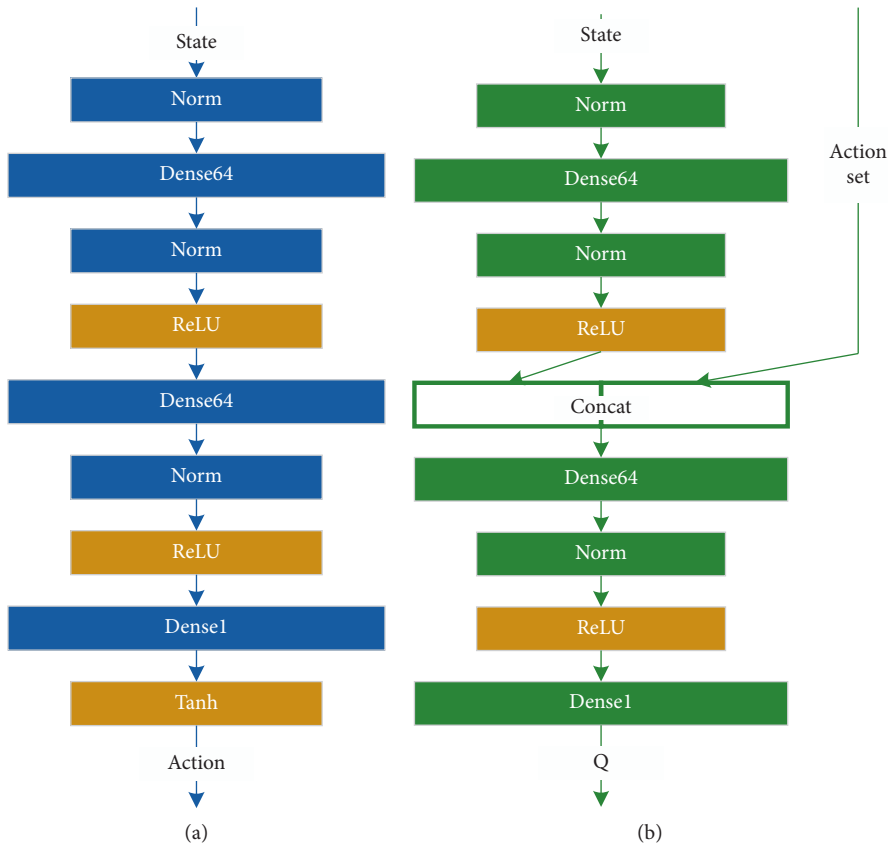


Figure 6: The network structures of actor and critic used in the proposed CoMADDPG. (a) Actor module. (b) Critic module.

Table 3: Hyperparameters of experiment.

Parameter	Value
Discounted factor γ	0.80
Minibatch size T	128
Soft update factor τ	0.998
Epoch U	300
Learning rate-actor	$10^{-4} \longrightarrow 0$
Learning rate-critic	$10^{-3} \longrightarrow 0$
Hidden layers number	2
Hidden units number	64
Optimizer	Adam

CoMADDPG will maintain an upward trend. In the evaluation stage, there is no collision in CoMADDPG, while DDPG still collides frequently.

Secondly, vehicle selection based on virtual lanes is an essential step in the transformation of reinforcement learning methods. To discuss the impact of different vehicle selection strategies on training, the experiment is designed to compare the virtual lane-based method with the actual lane-based method. Figure 9 exhibits the collision performance under virtual lane- and actual lane-based methods with the different number of vehicles considered. The virtual lane-based method is described in Section 3.3, and the actual lane-based method relies on the physical distance to select vehicles. In the experiment, the maximum number of the largest impact vehicles is set to 1, 3, and 6, respectively. Because one vehicle has conflicts with three of the four lanes, three is selected as a parameter. The influence of the front and rear vehicles is also considered here, so 6 is also used as a parameter for the experiment. 1 is set as a control parameter. As displayed in Figure 9, the actual lane-based method hardly achieves no collision, but as more vehicles are considered, the new number of collisions drops significantly. As for virtual lane-based method, it keeps fewer collisions. When considering that the number of vehicles reaches 6, the curve can reach a relatively horizontal state faster. This demonstrates the effectiveness of virtual lane-based vehicle selection in CoMADDPG to achieve no collision at unsignalized intersection.

Thirdly, to observe the influence of various lane lengths on average travel time, the experiment shows vehicles of different densities are running on lanes of different lengths and the average travel time is evaluated. Intuitively, a longer lane would allow vehicles to adjust their velocity to pass the intersection quickly. In Figure 10, there is no noticeable difference among different vehicles densities. The average travel time is proportional to the length of the lane, which means the proposed method can effectively deal with different lane lengths and vehicle densities. Moreover, low-density vehicles perform relatively poorly in long lanes. This is due to the sparse intervehicle spacing, resulting in insufficient coordination among vehicles.

Finally, the trained CoMADDPG is utilized for comparison with the optimization-based approach [18], which is

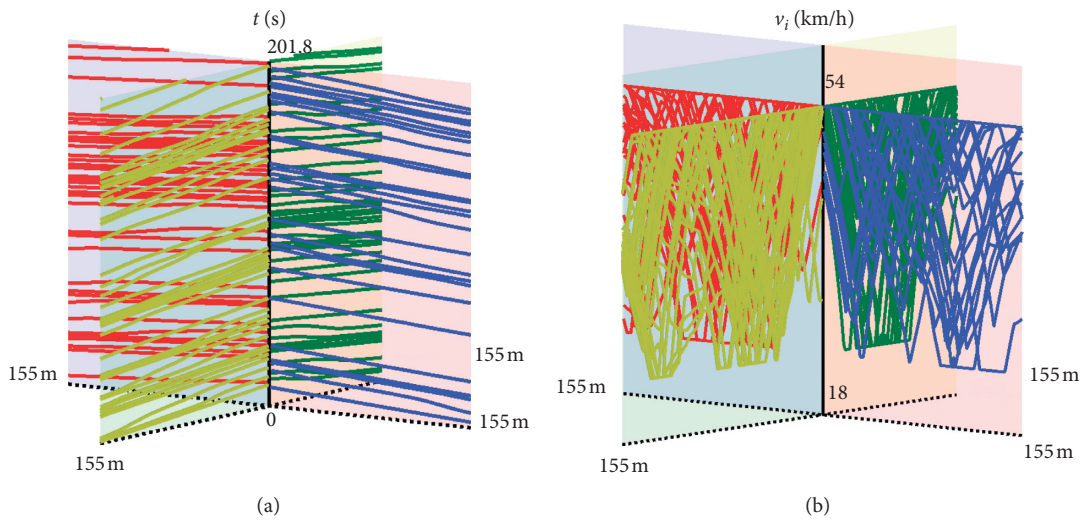


Figure 7: Vehicle status profiles in the area of intersection (different colours: vehicles from different entrances). (a) Position profiles. (b) Velocity profiles.

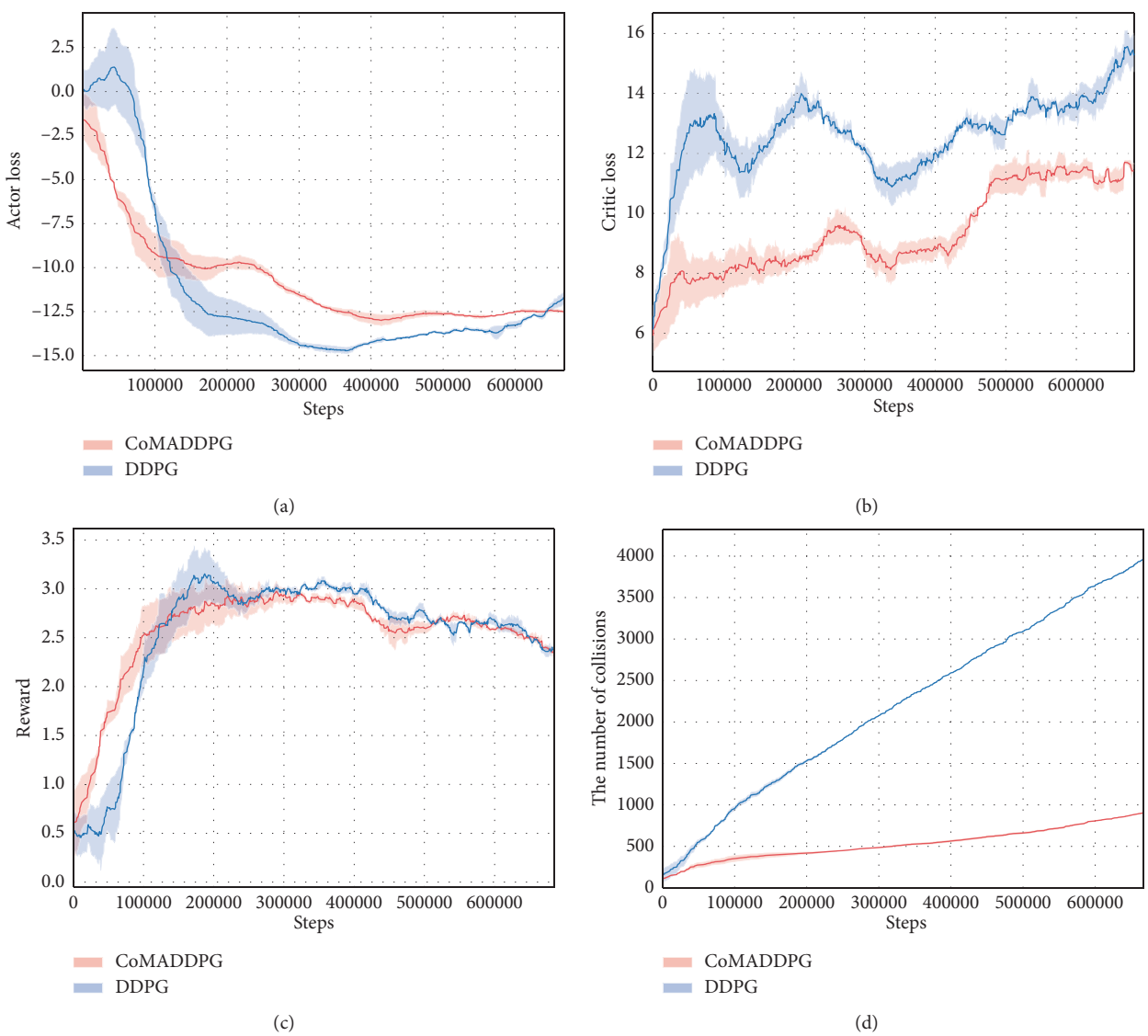


Figure 8: Comparison between CoMADDPG and DDPG during the training process. (a) The loss function of the actor. (b) The loss function of the critic. (c) The mean reward. (d) The cumulative number of collisions.

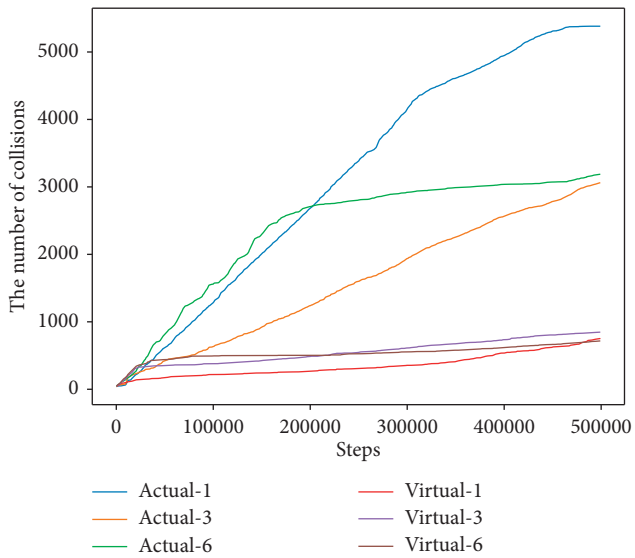


Figure 9: Comparison of cumulative collisions number using different vehicle selection configurations. "Actual" means actual lane-based vehicle selection. "Virtual" means virtual lane-based vehicle selection. The number indicates the maximum number of the largest impact vehicles.

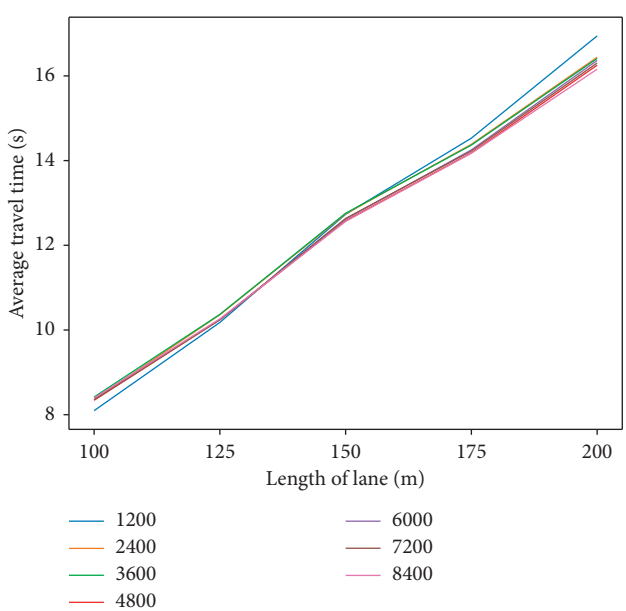


Figure 10: Comparison among various vehicles densities with different lengths of lane. 1200 to 8400 indicate different vehicles densities, whose unit is vehicles/hour.

in the same scenario. The average travel times in 3 different traffic volumes are shown in Table 4. It is observed that the proposed method increases the average travel times by 7.23%~39.28% in medium- and high-traffic volumes. Only in the case of low traffic flow the traffic efficiency is slightly lower than the optimization-based method. In addition, where more density is reached, the optimization-based method is no longer applicable, and the proposed CoMADDPG still works normally. Note that the results are only compared in the speed adjustment area, and the optimization-based method has an extra distance of 100 meters for observation and optimizing the speed. In brief, the results revealed that the proposed CoMADDPG has a slower travel time than the baseline, which can avoid vehicles congested in the entering lane of the intersection.

Table 4: Comparison of average travel times.

Total traffic volume (vehicles/hour)		1200	2400	3600	4800	6000
Travel time (s)	Ref. [18]	12.145	13.773	21.366	—	—
	Proposed	12.385	12.777	12.974	12.580	12.618
Improvement (%)		−1.98	7.23	39.28	—	—

The process of decision is evaluated on a laptop with an Intel CPU (i7-8565U @ 1.8 GHz, 1.9 GHz), 16 GB RAM, and NVIDIA GeForce MX250. The average running time is 0.36 ms. According to the 3 GPP standard [32], TX rate for cooperative collision avoidance between UEs supporting V2X applications is 100 messages/s, which means the message sending interval is 10 ms. Obviously, the proposed CoMADDPG meets the requirement of real time, and there is ample time to support function expansion.

5. Conclusion

In this paper, a multiagent reinforcement learning method is employed to solve distributed cooperation for connected and automated vehicles at unsignalized intersection, which has been regarded as a challenging problem of cooperation among dynamic quantity vehicles. Vehicle selection is incorporated into MADDPG to propose CoMADDPG, which makes it adapt to the dynamic quantity vehicles at the unsignalized intersection. Moreover, the virtual lane-based method enhanced intervehicle cooperation for collision avoidance. A typical 4-direction intersection containing four different types of vehicle is studied. The simulation results demonstrate that the proposed method is efficient. Compared with the existing optimization-based method, up to 39.28% improvement implies that CoMADDPG is worthwhile to handle distributed vehicle control safely and efficiently at unsignalized intersection.

In order to simplify the problem, this paper only studies the case where the single lane only goes straight. The proposed CoMADDPG can also solve the situation of multiple lanes and multiple directions. For multiple lanes, projecting more actual lanes onto virtual lanes requires an appropriate increase in the number of vehicles selected. For multiple directions, the piecewise projection of the collision points on the curve needs to be addressed.

In this paper, traffic safety and efficiency optimizations of passing through unsignalized intersection are researched, but vehicle stability is not considered. The introduction of vehicle stability would limit the exploration ability of the RL algorithm and may fall into the local optimum, which cannot achieve the highest vehicle passing efficiency. In future work, vehicle stability will guide our research as an essential topic.

Data Availability

No data were used to support this study. What we adopted is reinforcement learning, and data are generated from the environment we made in our simulation.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Key R&D Program of China (2016YFB0100902).

References

[1] S. Zeadally, J. Guerrero, and J. Contreras, "A tutorial survey on vehicle-to-vehicle communications," *Telecommunication Systems*, vol. 73, no. 3, pp. 469–489, 2020.

[2] L. Chen and C. Englund, "Cooperative intersection management: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, 2015.

[3] H. Ge, Y. Song, C. Wu, J. Ren, and G. Tan, "Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control," *IEEE Access*, vol. 7, pp. 40797–40809, 2019.

[4] D. Kim and O. Jeong, "Cooperative traffic signal control with traffic flow prediction in multi–intersection," *Sensors*, vol. 20, no. 1, p. 137, 2020.

[5] M. A. S. Kamal, T. Hayakawa, and J.–I. Imura, "Development and evaluation of an adaptive traffic signal control scheme under a mixed–automated traffic scenario," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, 2019.

[6] D. Yu, "Signal timing optimization based on fuzzy compromise programming for isolated signalized intersection," *Mathematical Problems in Engineering*, vol. 2016, Article ID 1682394, 12 pages, 2016.

[7] Y. Bai, J. Li, T. Li, L. Yang, and C. Lyu, "Traffic signal coordination for tramlines with passive priority strategy," *Mathematical Problems in Engineering*, vol. 2018, p. 14, 2018.

[8] Y. Xu, D. Li, and Y. Xi, "A game-based adaptive traffic signal control policy using the vehicle to infrastructure (V2I)," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9425–9437, 2019.

[9] Y. Lu, X. Xu, C. Ding, and G. Lu, "A speed control method at successive signalized intersections under connected vehicles environment," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 3, pp. 117–128, 2019.

[10] B. Xu, "Cooperative method of traffic signal optimization and speed control of connected vehicles at isolated intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1390–1403, 2018.

[11] S. Boudaakat, R. Ahmed, and B. Omar, "Smart traffic control system for decreasing traffic congestion," in *Proceedings of International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS)*, December 2019.

[12] Md A. S. Kamal, T. Hayakawa, and J.-I. Imura, "Development and evaluation of an adaptive traffic signal control scheme under a mixed–automated traffic scenario," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, 2019.

[13] S. Gupta, M. Vasardani, and S. Winter, "Negotiation between vehicles and pedestrians for the right of way at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 888–899, 2018.

[14] P. Dai, K. Liu, Q. Zhuge, E. H.-M. Sha, V. C. S. Lee, and S. H. Son, "Quality–of–experience–oriented autonomous intersection control in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1956–1967, 2016.

[15] Y. Guan, Y. Ren, S. E. Li et al., "Centralized Conflict–free Cooperation for Connected and Automated Vehicles at Intersections by Proximal Policy Optimization," 2019, https://arxiv.org/abs/1912.08410.

[16] B. Qian, H. Zhou, F. Lyu, J. Li, T. Ma, and F. Hou, "Toward collision-free and efficient coordination for automated vehicles at unsignalized intersection," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10408–10420, 2019.

[17] B. Xu, S. E. Li, Y. Bian et al., "Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 322–334, 2018.

[18] Y. Bian, S. E. Li, W. Ren, J. Wang, K. Li, and H. Liu, "Cooperation of multiple connected vehicles at unsignalized intersections: distributed observation, optimization, and control," *IEEE Transactions on Industrial Electronics*1 pages, 2019.

[19] A. Hadjigeorgiou and S. Timotheou, "Optimizing the trade–off between fuel consumption and travel time in an unsignalized autonomous intersection crossing," in *Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC)*, October 2019.

[20] F. Belkhouche, "Collaboration and optimal conflict resolution at an unsignalized intersection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2301–2312, 2018.

[21] Y.-C. Hsu, S. Gopalswamy, S. Saripalli, and D. A. Shell, "An MDP model of vehicle–pedestrian interaction at an unsignalized intersection," in *Proceedings of IEEE 88th Vehicular Technology Conference (VTC–Fall)*, IEEE, Chicago, IL, USA, Augest 2018.

[22] J. Wang, X. Zhao, and G. Yin, "Multi–objective optimal cooperative driving for connected and automated vehicles at non–signalised intersection," *IET Intelligent Transport Systems*, vol. 13, no. 1, pp. 79–89, 2018.

[23] H. Yang and K. Oguchi, "Intelligent vehicle control at signal–free intersection under mixed connected environment," *IET Intelligent Transport Systems*, vol. 14, no. 2, pp. 82–90, 2019, https://search.crossref.org/?q=Intelligent+vehicle+control+at+signal%E2%80%93free+intersection+under+mixed+connected+environment.+IET+Intelligent+Transport+Systems+14.2+%282019+%29%3A+.

[24] M. Kwon, J. Lee, and H. Park, "Intelligent IoT connectivity: deep reinforcement learning approach," *IEEE Sensors Journal*, vol. 20, no. 5, pp. 2782–2791, 2020.

[25] Y. Zhang, L. Guo, B. Gao, T. Qu, and H. Chen, "Deterministic promotion reinforcement learning applied to longitudinal velocity control for automated vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 338–348, 2019.

[26] V. Mnih, V. Kavukcuoglu, D. Silver et al., "Human–level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[27] T. P. Lillicrap, "Continuous control with deep reinforcement learning," 2015, https://arxiv.org/abs/1509.02971.

[28] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-

competitive environments," in *Proceedings of Advances in Neural Information Processing Systems. 31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, June 2017.

[29] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243–1253, 2019.

[30] M. Zhou, Yu Yang, and X. Qu, "Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: a reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, 2019.

[31] J. Lee, J. Chung, and K. Sohn, "Reinforcement learning for joint control of traffic signals in a transportation network," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, 2019.

[32] 3GPP TS22.186, "Enhancement of 3GPP support for 5G V2X services: stage 1," 2019.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2365076, 15 pages
https://doi.org/10.1155/2020/2365076

Research Article

DetReco: Object-Text Detection and Recognition Based on Deep Neural Network

Fan Zhang,[1,2] **Jiaxing Luan**,[1] **Zhichao Xu**,[1] **and Wei Chen**[1]

[1]*School of Electrical and Information Engineering, China University of Mining and Technology (Beijing), Beijing Ding No. 11, Xueyuan Road, 100083 Beijing, China*
[2]*Institute of Intelligent Mining and Robotics, China University of Mining and Technology (Beijing), Beijing Ding No.11, Xueyuan Road, 100083 Beijing, China*

Correspondence should be addressed to Fan Zhang; zf@cumtb.edu.cn

Received 13 April 2020; Accepted 12 June 2020; Published 14 July 2020

Guest Editor: Chi-Hua Chen

Deep learning-based object detection method has been applied in various fields, such as ITS (intelligent transportation systems) and ADS (autonomous driving systems). Meanwhile, text detection and recognition in different scenes have also attracted much attention and research effort. In this article, we propose a new object-text detection and recognition method termed "DetReco" to detect objects and texts and recognize the text contents. The proposed method is composed of object-text detection network and text recognition network. YOLOv3 is used as the algorithm for the object-text detection task and CRNN is employed to deal with the text recognition task. We combine the datasets of general objects and texts together to train the networks. At test time, the detection network detects various objects in an image. Then, the text images are passed to the text recognition network to derive the text contents. The experiments show that the proposed method achieves 78.3 mAP (mean Average Precision) for general objects and 72.8 AP (Average Precision) for texts in regard to detection performance. Furthermore, the proposed method is able to detect and recognize affine transformed or occluded texts with robustness. In addition, for the texts detected around general objects, the text contents can be used as the identifier to distinguish the object.

1. Introduction

Object detection [1, 2], as one of the most fundamental and challenging problems in computer vision, has received great attention in recent years. In the context of computer vision, object detection deals with the task of detecting instances of visual objects of specific classes such as humans, animals, and cars in digital images. It combines the cutting-edge technologies in many fields such as image processing, pattern recognition, automatic control, and artificial intelligence. Object detection is widely used in many fields including intelligent transportation systems [3, 4], advanced driver assistance systems (ADAS), and autonomous driving systems.

In intelligent traffic surveillance system [5], vehicle detection and recognition are a vital task. The automatic monitoring digital cameras take snapshots of passing vehicles and other moving objects to provide valuable clues including license plate number, the vehicle type, and the driver's facial image for authorities and other security departments. In recent years, autonomous cars and driverless vehicles have significantly changed the manner of transportation. Computer vision system is efficiently used in the development of ADAS. Sakhare et al. [6] have a detailed study of the vehicle detection in dynamic conditions. Yudin et al. [7] study vehicle detection in difficult areas with various architectures of deep neural networks [8].

In automated driving, detection and recognition of pedestrians, vehicles, traffic lights, and traffic signs [9] help avoid accidents and achieve safe driving. Collision avoidance systems are required for the driver to handle the emergence. Detecting pedestrians is essential for autonomous driving [10]. Zhang and Kim [11] propose a pedestrian detector which combines skip pooling from multiscale feature maps

and recurrent convolutional layers to detect pedestrians of different scales. Reliable traffic light detection and classification in urban environments are also crucial for automated driving [12, 13]. Kim et al. [12] develop a two-step method to detect traffic lights with SSD architecture. Lu et al. [14] utilize a visual attention model to detect traffic signals which is effective for the detection of small objects.

Object detection, which is the core of various intelligent transportation systems, has been a research hotspot in recent years. Meanwhile, the rapid development of deep learning has accelerated the development of object detection. Many deep learning based object detection techniques have led to giant breakthroughs and remarkable performance. Object detection can be divided into one-stage methods and two-stage methods. Object detection algorithm of two-stage methods usually involves two steps. Firstly, region proposals are obtained from the original image. Secondly, the classification and regression networks such as the R-CNN [15] (Regional Convolutional Neural Network) series are used to detect the region proposals. Object detection algorithm of one-stage method just needs one step. One-stage methods can accomplish the classification and bounding box regression tasks directly without finding the region proposals separately. Typical one-stage algorithms include SSD [16] (Single Shot Multibox Detector) and YOLO [17] (You Only Look Once).

R-CNN proposed by Ross B. Girshick uses selective search [18] method to perform ROI (Region of Interest) scaling and feature extraction on target images. Because R-CNN requires forward calculation for a large number of region candidates which may overlap each other, the speed of training and detection is very slow. Fast R-CNN [19] uses a feature extractor to extract the features of the entire image instead of extracting each image multiple times for each region proposal. Because Fast R-CNN does not extract features repeatedly, the processing time is significantly reduced. Faster R-CNN [20] uses a design similar to Fast R-CNN. Faster R-CNN replaces the selective search method with RPN (region proposal network), which solves the problem of excessive time overhead in generating ROI. The Faster R-CNN achieves high accuracy and detection speed to some extent, but it still cannot meet the real-time requirement.

Compared with Faster R-CNN, SSD has a significant advantage of detection speed. The network generates multiple feature maps at different scales. Then the classification and bounding box regression tasks are simultaneously done on multiscale feature maps. SSD is able to detect large objects effectively. YOLO is another one-stage method. It predicts bounding boxes and class probabilities of multiple objects simultaneously. However, different from the SSD algorithm, YOLO does not use multiscale feature maps for detection. Its generalization capability is poor for object with large scale variations compared with that of SSD. It leads to missed detection and low recognition accuracy. YOLOv2 [21] algorithm uses anchor mechanism which utilizes convolutional layers instead of fully connected layers as in YOLO to predict the bounding boxes. The disadvantage of using fully connected layer to predict bounding boxes is that the spatial information of feature map is lost. However, the anchor mechanism directly predicts the bounding boxes on the feature map with convolutional layers. The spatial information of feature map is well preserved. Each feature point of the feature map corresponds to each grid of the original image. YOLOv2 improves the performance of the detection accuracy. YOLOv3 [22] algorithm adopts multiscale feature maps to predict bounding boxes. YOLOv3 uses FPN (Feature Pyramid Networks) concept which uses the output of the middle layers to merge with the output of the latter layers. The high-level features are passed to the low layers, so that small objects on low-level feature maps can be better detected. YOLOv3 has been greatly improved in regard to detection speed and accuracy.

The majority of the recent works related to deep neural networks has been devoted to detection or classification of object categories [23]. On top of that, another problem in computer vision that plays a vital role in intelligent transportation systems is the image-based text recognition. Text recognition aims to decode a sequence of labels from cropped text images.

The conventional methods recognize the text contents at character level. The characters of the text are segmented from the cropped text image. Then the segmented character regions are preprocessed and recognized. Different from the character-level recognition methods, recent text recognition methods do not require character segmentation in advance. One famous method is the multidigit number classification proposed by Goodfellow et al. [24], which is based on DCNN (deep convolutional neural network). The method requires selecting the maximum predictable sequence length in advance. This limits it to recognizing house number or license plate number whose length of texts is known beforehand. Another commonly used method is RNN (recurrent neural network) with CTC [25] (connectionist temporal classification). Shi et al. [26] and He et al. [27] propose RNN models to encode the features from the CNN and adopt CTC to decode the encoded sequence. The advantage of this method is that it can generate texts of any length. Furthermore, the nature of the Recurrent Neural Network determines that the model is able to learn the relationship between text and text temporal relations. Another type of method that does not require character segmentation of texts is attention mechanism. Lee and Osindero [28] use attention-based sequence-to-sequence structure to automatically focus on certain extracted CNN features and directly use text images to perform word string learning. This method implicitly learns character-level language models embodied in RNN. It is able to perform text recognition in unconstrained natural scenes.

Scene text recognition [29] in intelligent transportation systems has many applications, such as vehicle license plate recognition and road sign recognition. As an important part of intelligent transportation systems, vehicle license plate recognition is widely used in intelligent monitoring systems and parking systems. Automatic license plate recognition (ALPR) refers to the extraction of vehicle license plate information from an image or a sequence of images [30]. Chai

and Zuo [31] propose an automatic vehicle license plate recognition method which adopts edge detection algorithm in extraction and character segmentation and recognition. Chang et al. [32] use license plate recognition technology to track vehicle on the road in complex traffic conditions.

Object detection in applications refers to the detection under specific application scenarios, such as pedestrian detection, vehicle detection, and scene text detection. Text recognition in specific application scenarios can get more information from the objects on which the applications focus. In this paper, we propose a model which combines object-text detection and text recognition. The model is able to detect both texts and general objects simultaneously. The model combines object detection task and text detection task and recognizes the detected text contents. In addition, for the texts detected around general objects, the contents can be used as the identifier to distinguish the object. The method we propose can be applied to a wide range of applications in regard to intelligent transportation systems and has comprehensive capabilities of detection and recognition.

The contributions are summarized as follows:

(1) We propose an object-text detection model for multiple objects which can simultaneously detect texts and general objects

(2) We propose a text recognition framework that effectively combines text detection and recognition

(3) The method we propose can detect multiple types of objects and instantiate the identities of the detected objects based on the identified text labels. The recognized text label is used as a valid identity of the object

2. Materials and Methods

The network structure in this paper consists of two parts: object-text detection network and the text recognition network. We use the YOLOv3 architecture which adopts a fully convolutional neural network [33] to detect objects and texts in real-scene images. The convolutional network is used to extract the features in multiple scales feature maps from the image. The classification and bounding box regression networks directly output the objectness score, the class of the object, and the coordinate offsets of the object at multiple feature maps. We use NMS [34] (nonmaximum suppression) to remove the redundant bounding boxes which have large overlap with the same object. We adopt a successful scene text recognition algorithm, CRNN [26] (Convolutional Recurrent Neural Network), in conjunction with object-text detection. According to the coordinates of the text type which are output from the object-text detection network, the text regions are cropped from the original image. A convolutional neural network is used to extract features from the text regions. The extracted feature maps need to be scaled to a uniform height with a fixed aspect ratio. We use the recurrent model to encode the feature sequences from the feature maps and CTC to decode the encoded sequence. The network structure we propose is shown in Figure 1.

2.1. Architecture of the Object-Text Detection Network. The backbone network adopts Darknet-53, which uses the former 52 layers without fully connected layer. The feature extraction network is a fully convolutional network. It is mainly composed of 3×3 and 1×1 convolution kernels and a large number of shortcut links with residual units [35–37]. The structure of the feature extraction network is shown in Figure 2. The network uses convolution kernel with stride instead of pooling layers to reduce the negative gradient effects brought by pooling. We also adopt a lot of data augmentation and batch normalization to avoid overfitting. In order to enhance the accuracy of the algorithm for small object detection, the network adopts upsampling and fusion methods which are similar to FPN [38] to implement the multiscale feature maps.

As shown in Figure 3, we assume the size of the input image to be 416×416. We extract three different scale feature maps from 26th, 43rd, and 52nd layers of the feature extraction networks in Figure 2. The scales of the extracted feature maps are 13×13, 26×26, and 52×52. The feature fusion network outputs three different scale feature maps with upsampling and fusion. The top layer with a size of 13×13 is concatenated with the 26×26 feature map via one-time upsampling. Then it is concatenated with the 52×52 feature map by upsampling twice. In this way, the high-level features from the top layer are passed to the low layers, which makes the model better at detecting small objects on low-level feature maps. Finally, the network generates three feature maps of different scales which are 1/8, 1/16, and 1/32 of the original image.

The output layers in 3 different scales are also convolutional. In our experiments with our dataset which has twenty-one classes including twenty general categories and one text category, we predict 3 bounding boxes with different sizes at feature maps of each scale. The shape of the output tensor is $N \times N[3 \times (4 + 1 + 21)]$, where N is the scale of the feature map, 3 is the anchor boxes in features of different scales, 4 is the coordinate offsets of the bounding box, 1 is the objectness confidence prediction, and 21 is the object classes.

The network adopts the anchor-based mechanism. Each gird of the feature maps predicts 3 bounding boxes according to the anchor boxes of 3 different scales. There are in total 9 different scale anchor boxes which are generated from k-means clustering. The 9 clusters on the COCO dataset [39] are (10×13), (16×30), (33×23), (30×61), (62×45), (59×119), (116×90), (156×198), and (373×326). The anchor boxes in different scale feature maps are shown in Figure 4.

The object-text detection network simultaneously predicts bounding boxes of texts and general objects conditioned on its input feature maps. At each grid of associated feature map, it outputs the objectness confidence, classification score, and coordinate offsets to its associated anchor boxes in a convolutional manner.

The object-text detection network adopts logistic regression to predict the bounding boxes and the objectness scoring on each anchor. Only the anchor with the highest objectness score is calculated. Each object can be detected by

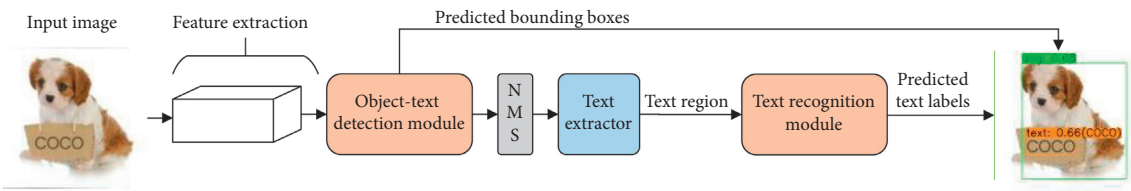


Figure 1: Overall architecture of the proposed network architecture. Feature maps are firstly extracted with convolutional layers. The object-text detection module is built on top of the feature maps to predict the bounding boxes of the texts and general objects. The NMS module is used to remove the reductant bounding boxes and retain the final positive bounding boxes. The text extractor extracts the text regions corresponding to the coordinates of the text bounding boxes which are the output of the object-text detection module. The text regions are then fed into the text recognition module.

only one anchor. This step is performed before prediction, which can remove unnecessary anchors and reduce the amount of calculation. In bounding box regression, the network outputs the coordinate offsets. The formula that converts offsets to bounding box coordinates is defined as

$$\begin{aligned} b_x &= \sigma(t_x) + c_x, \\ b_y &= \sigma(t_y) + c_y, \\ b_w &= p_w e^{t_w}, \\ b_h &= p_h e^{t_h}, \end{aligned} \tag{1}$$

where $b_x, b_y, b_w,$ and b_h are the coordinates of the bounding box, $c_x, c_y, p_w,$ and p_h are the coordinates of the anchor box, and $\sigma(\cdot)$ represents the sigmoid function.

2.2. Loss Function of the Object-Text Detection Network. Objectness confidence is the probability of predicting the existence of the object-text in anchor box. Objectness confidence loss adopts binary cross entropy. The objectness confidence loss function is defined as

$$\begin{aligned} L_{\text{conf}}(o, c) &= -\sum\left(o_i \ln(\hat{c}_i) + (1 - o_i)\ln(1 - \hat{c}_i)\right), \\ \hat{c}_i &= \text{Sigmoid}(c_i), \end{aligned} \tag{2}$$

where $o_i \in \{0, 1\}$ represents the existence of the object-text in anchor box and $\hat{c}_i$ represents the sigmoid probability of the existence of the bounding box.

Object-text classification score is the probability of the class which the object-text belongs to. The object-text class loss function is defined as

$$\begin{aligned} L_{cls}(O, C) &= -\sum_{i \in Pos} \sum_{j \in cls} \left(O_{ij} \ln\left(\widehat{C_{ij}}\right) + \left(1 - O_{ij}\right)\ln\left(1 - \widehat{C_{ij}}\right)\right), \\ \widehat{C_{ij}} &= \text{Sigmoid}\left(C_{ij}\right), \end{aligned} \tag{3}$$

where $O_{ij} \in \{0, 1\}$ represents the existence of the object-texts' class j in anchor box i and $\widehat{C_{ij}}$ represents the sigmoid probability of the class j of the bounding box i.

Object-text detection model predicts the coordinate offsets between anchor boxes and bounding boxes. Equation (1) is used to convert the offsets to the coordinates of the bounding box. The object-text location loss adopts the GIoU [40] (Generalized Intersection over Union) method to calculate the error between the bounding box and ground truth. The GIoU Loss Algorithm is defined as in Algorithm 1.

Layers	Filters			Feature maps	
Convolutional	32	3 ∗ 3		416 ∗ 416	
Convolutional	64	3 ∗ 3	stride = 2	208 ∗ 208	
Convolutional	32	1 ∗ 1			
Convolutional	64	3 ∗ 3			×1
Residual				208 ∗ 208	
Convolutional	128	3 ∗ 3	stride = 2	104 ∗ 104	
Convolutional	64	1 ∗ 1			
Convolutional	128	3 ∗ 3			×2
Residual				104 ∗ 104	
Convolutional	256	3 ∗ 3	stride = 2	52 ∗ 52	
Convolutional	128	1 ∗ 1			
Convolutional	256	3 ∗ 3			×8
Residual				52 ∗ 52	
Convolutional	512	3 ∗ 3	stride = 2	26 ∗ 26	
Convolutional	256	1 ∗ 1			
Convolutional	512	3 ∗ 3			×8
Residual				26 ∗ 26	
Convolutional	1024	3 ∗ 3	stride = 2	13 ∗ 13	
Convolutional	512	1 ∗ 1			
Convolutional	1024	3 ∗ 3			×4
Residual				13 ∗ 13	

Figure 2: The feature extraction network. The backbone network of detection model uses the former 52 layers of the Darknet-53 without fully connected layer to extract features.

We use $L_{\text{loc}} = \text{GIoU\_Loss}$ to form of the object-text location loss. The total loss function can be represented as

$$L_{\text{tol}} = \alpha L_{\text{conf}} + \beta L_{\text{cls}} + \gamma L_{\text{loc}}, \tag{4}$$

where α, β, and γ are the weights of each loss. We empirically set $\alpha = \beta = \gamma = 1$.

2.3. NMS Module. The NMS module is applied to remove the redundant object-text bounding boxes detected from the same object. We adopt the NMS after the object-text detection on the object-text bounding boxes.

2.4. Text Recognition. After the object positions are detected from the object-text detection network, we pick out the text-type bounding boxes based on the text class. Firstly, the text

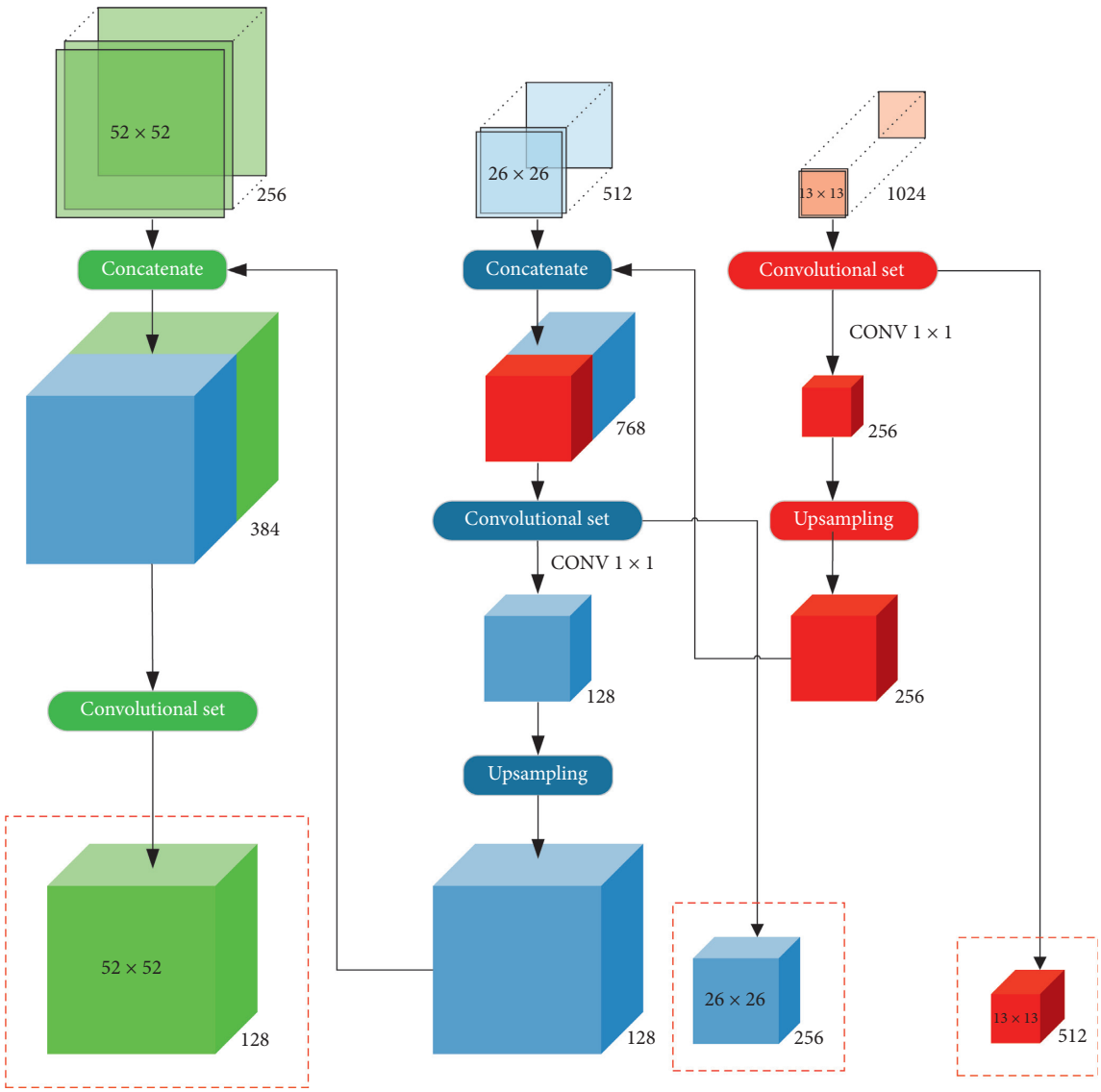


Figure 3: The feature fusion network. The generation of the 3 different scale feature maps uses upsampling and concatenation from convolutional layers. Firstly, the 3 different scale feature maps are extracted from the Darknet-53, respectively, in different convolutional layers. Secondly, the convolutional set which consists of a set of convolutional layers is used to reduce the channel of the 13×13 feature map. Then the 13×13 feature map is upsampled with stride 2 to concatenate with the 26×26 feature map. The 52×52 feature map concatenation process is similar to that of the 13×13 feature map. Finally, the red dashed boxes are the extracted feature maps in 3 different scales.

extractor extracts the text regions corresponding to the coordinates of the text bounding boxes produced by the object-text detection module. Then the text recognition module preprocesses the extracted text regions by resizing them before they are fed into convolutional neural network. We scale the text regions to $(32, 100, 3)$ with a fixed aspect ratio, where 32 is the fixed height, 100 is the maximum length, and 3 represents the number of the image channel. Finally, we use the scaled text region as the input of the convolutional layers.

We adopt the CRNN model as our text recognizer. Firstly, the convolutional layers extract the feature maps from the preprocessed text region. A sequence of feature vectors is extracted from left to right from the feature maps. Then each frame of the sequences which represents a vertical region corresponding to the original text image becomes the input of the recurrent layers. The recurrent layers adopt the deep bidirectional LSTM [41] (long short-term memory) to encode the sequence of the feature vectors. Finally, we adopt CTC to predict the text label corresponding to the sequences from the recurrent layers.

3. Results and Discussion

3.1. Experiment Setup. The object-text detection network is trained with training images using Adam (adaptive moment estimation) [42]. We initialize the model with pre-trained weights on the COCO dataset. We divide the training process into two stages. In the first stage, we fix the backbone network and just train the classification and

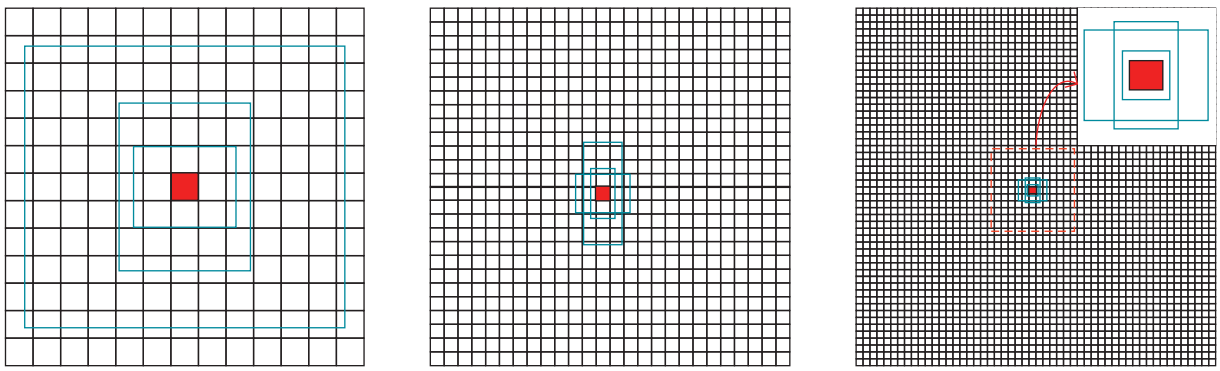


Figure 4: The anchor boxes in 3 scale feature maps.

regression network. In the second stage, we train the whole network.

When we train the object-text detection network, in the first two epochs in training, we adopt the method of gradually increasing the learning rate from low to high which is called "warmup stage" method. The network converges quickly with the large learning rate. And then, we need to stabilize the network with a low learning rate to avoid gradient oscillation. We adopt a cosine annealing strategy proposed by Loshchilov et al. [43]. At the i-th training step, the learning rate decays with a cosine annealing as follows:

$$\eta = \eta_{\text{end}} + 0.5\left(\eta_{\text{init}} - \eta_{\text{end}}\right)\left(1 + \cos\left(\frac{T_{\text{cur}} - T_{\text{warm}}}{T_{\text{train}} - T_{\text{warm}}}\right)\pi\right), \tag{5}$$

where η_{init} is the initial value of the learning rate, which is set to 10^{-4}, η_{end} is the end value which we set to 10^{-6}, and T_{cur} accounts for how many steps have been performed. T_{train} is the total steps during the training. T_{warm} represents the warmup steps in the first two epochs. The learning rate curve is shown in Figure 5.

The training algorithm of the object-text detection model is summarized as in Algorithm 2.

We use a CRNN model proposed by Shi et al. [26] as the text recognition network. The experiment uses a pretrained model trained on the synth90k dataset [44] to initialize the parameters of the text recognition model. We use NEOCR [45] dataset and SCUT FORU dataset to fine-tune the pretrained model. We set the training parameters as follows: The model training runs for 2000000 epochs. The batch size is 32. The initial learning rate is 0.01 with exponential decay of 0.1 every 500000 epochs. The experiment adopts gradient descent with momentum [46] to train the text recognition network. We set the parameter of momentum to 0.9.

The training algorithm of the text recognition model is summarized as in Algorithm 3.

3.2. Dataset. We evaluate the proposed method on four datasets: VOC 2007 [47], VOC2012 [48], ICDAR 2013 [49], and SCUT FORU DB. VOC2007 and VOC2012 are the datasets about object detection. ICDAR 2013 and SCUT FORU DB are the datasets about text detection. We integrate them into a comprehensive dataset for detecting type-text object and general object simultaneously.

VOC2007 is the challenge to recognize objects from a number of visual object classes in realistic scenes. The database contains a total of 9963 annotated images. We use 5011 images as training set and 4952 images as testing set. There are twenty object classes in the dataset.

VOC2012 is the same challenge as VOC2007 which increases the size of the training set. There are 17125 training images in total. The testing set has not been released yet.

ICDAR 2013 is the Challenge 2 of ICDAR 2013 Robust Reading Competition, which contains horizontal texts. The dataset focuses on the reading of texts in real scenes. The images of the dataset refer to the text images focused around the text content of interest. The dataset consists of 229 training images and 233 testing images. Due to the fact that there are too few training images, we additionally use 1200 images from SCUT FORU training dataset.

SCUT FORU Database is released by the South China University of Technology. The dataset consists of Chinese2k and English2k.We only use the English2k dataset. The English2k dataset contains character annotations and word annotations. The characters of the dataset include 52 upper-lowercase letters and 10 Arabic numerals. The label format of the dataset is $\{x, y, w, h, \text{label}\}$. $\{x, y\}$ are the top-left coordinates of the rectangular box. $\{w, h\}$ are the width and height of the rectangular box. {label} is the word label of the text region. There are a total of 1715 images, of which 1200 are the training images and 515 are the testing images. The dataset has an average of 18.4 characters and 3.2 words per image.

COCO Dataset is a large-scale dataset for object detection, segmentation, and captioning. It contains more than 330K images and 200K labels. The COCO dataset has 80 object categories in total.

In the experiment, we integrate the datasets into a comprehensive dataset of 29265 images in total. There are 23565 training images and 5700 testing images. Since these datasets have different annotation formats, we need to convert them into a unified annotation format. The coordinates format of the annotation is defined as

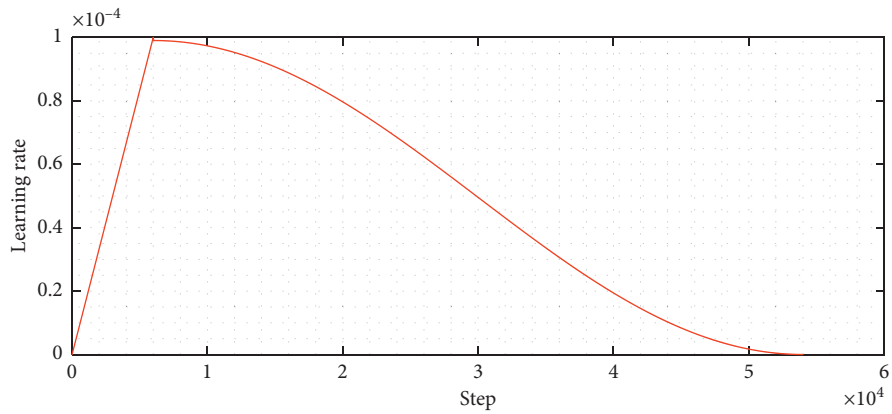


Figure 5: Learning rate schedule.

$\{x_{\min}, y_{\min}, x_{\max}, y_{\max}\}$. We shuffle the combined dataset to feed into the model.

Text recognition network is performed with a CRNN model proposed by Shi et al. We use a pretrained model trained on the synth90k dataset and use NEOCR dataset and SCUT FORU dataset to fine-tune the pretrained model. The annotations in NEOCR dataset contain characters that are not in the English alphabet. We have modified the annotations by replacing the special characters to English letters that look similar. The text images in SCUT FORU dataset are cropped from the original images corresponding to the coordinates in annotations. The text images are resized to the size of (32×100) before they are fed into the text recognition network.

3.3. Evaluation Metrics. We use mAP (mean Average Precision) as the measurement to evaluate the detection model performance. The mAP calculation is based on the following metrics [50]:

True Positives (TP): examples detected correctly with IoU ≥ threshold

False Positives (FP): negative examples detected incorrectly with IoU < threshold

False Negatives (FN): the ground truths not detected

The threshold is usually set to 0.5, 0.75, or 0.95. In our evaluation, we set it to 0.5.

Precision. Precision is the percentage of correct positive predictions. The precision is defined as

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{6}$$

Recall. Recall is the percentage of true positive detected among all relevant ground truths. The recall is defined as

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{7}$$

PR (Precision-Recall) Curve. The PR curve is a good way to evaluate the performance of an object detector. The precision and recall values of detected objects are plotted to get a PR curve. The area under the PR curve is called AP (Average Precision). The AP calculation is defined as

$$\text{AP} = \int P(R)\text{d}R, \tag{8}$$

where $P(R)$ is the measured precision against recall.

mAP. The mAP is the average of all categories of AP.

3.4. Analysis of Experimental Results. In order to verify the choice of YOLOv3 as the detection network in the proposed method, we compare the detection performance of different detection frameworks, namely, Fast R-CNN, Faster R-CNN, SSD, YOLO, YOLOv2, and YOLOv3. We as well compare different input size setups of YOLOv3. The results are shown in Table 1. All the detection frameworks being compared are trained on the VOC2007 and VOC2012 training datasets and the mAP is tested on the VOC2007 testing dataset. As can be seen in Table 1, the YOLOv3 framework with network input size of 416×416 achieves the highest mAP among the frameworks being tested. Further, the YOLO series, be it YOLOv2 or YOLOv3, generally achieve higher mAP than other frameworks. It can therefore be concluded that the choice of the YOLOv3 framework in the proposed method is an optimized solution.

After we have confirmed the performance of the YOLOv3 in object detection, we further train it on the comprehensive dataset which is composed of the general object detection datasets of VOC2007 and VOC2012 and the text detection datasets of SCUT FORU and ICDAR2013. Then we test the performance of the frameworks on different testing datasets. The general objects detection testing dataset VOC2007 and the text detection datasets SCUT FORU and ICDAR2013 are used. We compare the performance of different detection frameworks with 3 categories out of the total 20 categories in the PASCAL VOC 2007 dataset. As shown in Table 2, the performance of YOLOv3 on the 3 categories is much better than other detection frameworks. We verify that the YOLOv3 has excellent performance on object detection. Our model achieves 70.0 mAP in the text

Input: The region of the GT and BB (GT, $BB \subseteq S \in \mathbb{R}^n$), where S is the input image size.
Output: *GIoU Loss*
Step 1. Calculate the smallest enclosing region $C, C \subseteq S \in \mathbb{R}^n$;
Step 2. $IoU = |GT \cap BB|/|GT \cup BB|$;
Step 3. $GIoU = IoU - (|C - (GT \cup BB)|/|C|)$;
Step 4. Calculate the bounding box scale: $gt\_scale = 2 - (GT/S)$;
Step 5. Calculate the location loss function: $GIoU\_Loss = O \cdot gt\_scale \cdot (1 - GIoU)$, where O is the existence of the object-text in associated bounding box.

Algorithm 1: GIoU_Loss algorithm.

Input: Parameter_1: The training set $X = \{x_1, x_2, x_3, \ldots, x_i, \ldots x_N\}$. N is the number of the batches. x_i is the i-th batch of the training set;
Parameter_2: The labels corresponding to the training images of which format is defined as: $\{x_{\min}, y_{\min}, x_{\max}, y_{\max}, class\}$
Output: Weights of the model
for $epoch\ in\ epochs$ **do**
  if $epoch \leq first\_stage\_epochs$ **then**
    $initialize\_train\_weights \longleftarrow classification\_regression\_weights$;
  else
    $initialize\_train\_weights \longleftarrow backbone\_weights + classification\_regression\_weights$;
  for $batch\ in\ batches$ **do**
    Predict the offsets, objectness and class:
    $x_i, y_i \longleftarrow preprocess(Input)$;
    $\hat{y}_i \longleftarrow forward(W, x_i)$;
    Calculate the loss:
    $l \longleftarrow loss(\hat{y}_i, y_i)$;
    Calculate the gradients:
    $grad \longleftarrow backward(l)$;
    Update the model parameters:
    $W \longleftarrow update(grad)$;
    end for
end for

Algorithm 2: Detection network training algorithm.

detection task. We are not listing the text detection performance of the other methods because they do not feature text detection and recognition.

One may notice that the mAP of YOLOv3 in Table 2 is lower than that of Table 1. This is because we further train the YOLOv3 network on the text detection datasets. The detection of text objects reduces the mAP to a certain extent. In addition, the text object in datasets of VOC2007 and VOC2012 is not marked in the annotations. The detected texts in VOC2007 and VOC2012 will be seen as 'False Positives', thus the mAP would decrease.

Due to the imperfection of the comprehensive dataset which consists of general object datasets and text datasets, we improve the annotation information of the comprehensive dataset. We label the text objects in VOC2007 and VOC2012 and the general objects in ICDAR2013 and SCUT. This makes the comprehensive dataset of the object detection more accurate, reduces the false positive rate of the detection model in training and testing, and improves the detection accuracy as a whole. As can be seen from Table 2, the detection model used in the experiment has the highest detection accuracy on the YOLOv3 framework with the size of 544×544. Table 3 compares the detection effect of the comprehensive dataset before and after the modification on the YOLOv3 framework with the size of 544×544. As shown in Table 3, the detection network on modified comprehensive dataset has higher accuracy on person and text objects than original dataset. The detection accuracy of the text object is significantly improved. The mAP on the modified comprehensive dataset has also improved.

3.5. Performance on Object-Text Detection and Recognition. The model we propose performs two tasks: object-text detection and text recognition. The object-text detection network can detect general objects and text objects simultaneously. The text contents of the detected text regions from the detection network are recognized by the text recognition network. This section shows the detection and recognition results of test images in the experiment.

As shown in Figure 6, we mainly show some detection results of test images in transportation. The detection model can detect multiple objects in one image. It has good performance on both small objects and large objects. The text

Input: Parameter_1: The training set of text $X = \{x_1, x_2, x_3, \ldots, x_i, \ldots x_N\}$. N is the number of the batches. x_i is the i-th batch of the training set;
Parameter_2: The text labels corresponding to the text training images: $\{label\}$
Output: Weights of the model
for $epoch\ in\ epochs$ **do**
  $initialize\_weights \longleftarrow pretrained\_weights$;
  for $batch\ in\ batches$ **do**
    Predict the recognition result:
    $x_i, y_i \longleftarrow preprocess(Input)$;
    $\hat{y}_i \longleftarrow forward(W, x_i)$;
    Calculate the CTC loss:
    $l \longleftarrow ctc\_loss(\hat{y}_i, y_i)$;
    Calculate the gradients:
    $grad \longleftarrow backward(l)$;
    Update the model parameters:
    $W \longleftarrow update(grad)$;
    end for
end for

Algorithm 3: Recognition network training algorithm.

Figure 6: Examples of the object detection result on PACSAL VOC2007. We use rectangular boxes to represent the location of the detected objects. The different colours of the rectangular boxes represent the different categories.

detection dataset contains billboards, signboards, road sign, etc. Some texts exist in complex environments and they might be occluded. As shown in Figure 7, the detection model can detect the text in complex scenes. However, some text bounding boxes in images are not accurate enough, which may cause wrong recognition in texts. The object-text detection model we propose can simultaneously detect the text and general objects. Some detection examples are demonstrated in Figure 8.

The text recognition model can recognize the text contents of the text regions detected from the detection model. As shown in Figure 9, we demonstrate some examples of text recognition model on road sign. As shown in Figure 10, the text recognition model can recognize not only the horizontal text, but also the affine distorted text. The affine distorted texts exist commonly due to the variations of the camera views. Yet the proposed model not only locates these texts, but also finds the contents of the texts.

Figure 11 gives a more application specific demonstration of the proposed object detection and text recognition model. In this scenario, information extracted by the text recognition module identifies the detected object. We use some cars images with plates as the proof of concept. The object-text detection model we have proposed can simultaneously detect the car and the plates on the car. Then the text recognition model recognizes the text contents on the plates.

Figure 7: Examples of the text detection result on SCUT FORU dataset. We use yellow solid boxes with overstriking to represent the location of the detected text.

Figure 8: Examples of object-text detection on the comprehensive dataset. The different colours of the rectangular boxes represent the different categories. The red dashed rectangular boxes represent the misdetections.

Figure 9: Examples of word spotting result. The yellow solid boxes with overstriking are the text locations. The words in parenthesis are the word recognition results. The red dashed boxes are the wrong word recognition.

Figure 10: Recognition examples of text with affine transforms. The yellow solid boxes with overstriking are the text locations. The words in yellow are the word recognition results. The red solid boxes are the misdetections.

Figure 11: Examples of object-text detection and recognition results. There are some images of car with plate. We use yellow box to represent the detection results of car. The yellow solid boxes with overstriking are the text locations. The words in yellow are the word recognition results.

Table 1: Detection performance on the PASCAL VOC 2007 testing dataset of different frameworks.

Detection frameworks	Trained on	mAP
Fast R-CNN [19]	VOC2007 + VOC2012	70.0
Faster R-CNN VGG-16 [20]	VOC2007 + VOC2012	73.2
Faster R-CNN ResNet [35]	VOC2007 + VOC2012	76.4
YOLO [17]	VOC2007 + VOC2012	63.4
SSD300 [16]	VOC2007 + VOC2012	74.3
SSD512 [16]	VOC2007 + VOC2012	76.8
YOLOv2 544×544 [21]	VOC2007 + VOC2012	78.6
YOLOv3 416×416	VOC2007 + VOC2012	**87.4**
YOLOv3 544×544	VOC2007 + VOC2012	86.8
YOLOv3 608×608	VOC2007 + VOC2012	86.1

Table 2: Performance of different detection frameworks on different testing datasets.

Detection frameworks	VOC2007				SCUT_FORU	ICDAR2013
	mAP	Car	Bus	Person	Text	Text
Fast R-CNN [19]	70.0	78.6	81.6	69.9	—	—
Faster R-CNN VGG-16 [20]	73.2	84.7	83.1	76.7	—	—
SSD300 [16]	74.3	84.2	83.0	76.2	—	—
SSD512 [16]	76.8	87.5	86.2	79.7	—	—
YOLOv3 416×416	77.9	90.4	85.4	85.7	70.0	64.2
YOLOv3 544×544	**78.3**	**91.5**	**89.4**	**86.8**	**70.0**	**70.0**
YOLOv3 608×608	77.9	90.4	85.4	85.7	70.0	64.2

Table 3: Detection performance between the comprehensive dataset and modified comprehensive dataset.

Detection frameworks	Dataset	mAP	Car	Bus	Person	Text
YOLOv3 544×544	Comprehensive dataset	76.9	**89.3**	**87.9**	83.5	66.0
	Comprehensive dataset (modified)	**77.2**	88.5	85.3	**83.9**	**72.8**

4. Conclusions

We present an object-text detection and recognition model in this article. The model not only detects the texts and general objects simultaneously but also recognizes the text contents inside the detected text bounding boxes. The method we have proposed combines both object detection and text recognition. In the applications of some scenarios, the recognized text contexts around the general objects are able to be used as the identifier to distinguish the object. The proposed method has potential in extensive applications, such as intelligent transportation systems and autonomous driving.

Possible directions for future research include the following:

(1) Improving the dataset: this refers to adding more samples which contain both text and general object to train the network

(2) Improving the detection network on the text detection: for example, the anchor box which is suitable for the text size can be used. We can use k-means to cluster on the dataset containing text objects to make the size of the generated anchor boxes more suitable for text

(3) Optimizing the connection between the detection network and the recognition network: in our proposed model, the connection between detection and recognition network is the text region which is cropped from the original image corresponding to the coordinates of the detected text boxes. In order to optimize the connection, we can extract the feature map from the detection network as the input of the recognition network. The affine transformation is applied to the feature map extracted from detection network to fit the input size of recognition network. Thus, during backpropagation, the gradients can flow from the recognition network back to the detection network. The detection and recognition model can be regarded as an end-to-end model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the Foundation of the National Key Research and Development Program (grant number 2016YFC0801800), National Natural Science Foundation of China (grant number 51874300), National Natural Science Foundation of China and Shanxi Provincial People's Government Jointly Funded Project of China for Coal Base and Low Carbon (grant number U1510115), and the Open Research Fund of Key Laboratory of Wireless Sensor Network & Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences (grant numbers 20190902 and 20190913).

References

[1] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: a survey," pp. 1–40, 2019, https://arxiv.org/pdf/1905.05055.

[2] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[3] C.-H. Chen, "A cell probe-based method for vehicle speed estimation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E103.A, no. 1, pp. 265–267, 2020.

[4] C.-H. Chen, F.-J. Hwang, and H.-Y. Kung, "Travel time prediction system based on data clustering for waste collection vehicles," *IEICE Transactions on Information and Systems*, vol. E102.D, no. 7, pp. 1374–1383, 2019.

[5] Y. Tang, C. Zhang, R. Gu, P. Li, and B. Yang, "Vehicle detection and recognition for intelligent traffic surveillance system," *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 5817–5832, 2017.

[6] K. V. Sakhare, T. Tewari, and V. Vyas, "Review of vehicle detection systems in advanced driver assistant systems," *Archives of Computational Methods in Engineering*, vol. 27, no. 2, pp. 591–610, 2020.

[7] D. A. Yudin, A. Skrynnik, A. Krishtopik, I. Belkin, and A. I. Panov, "Object detection with deep neural networks for reinforcement learning in the task of autonomous vehicles path planning at the intersection," *Optical Memory and Neural Networks*, vol. 28, no. 4, pp. 283–295, 2019.

[8] C.-H. Chen, F. Song, F.-J. Hwang, and L. Wu, "A probability density function generator based on neural networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 541, p. 123344, 2020.

[9] J. Zhang, M. Huang, X. Jin, and X. Li, "A real-time Chinese traffic sign detection algorithm based on modified YOLOv2," *Algorithms*, vol. 10, no. 4, p. 127, 2017.

[10] H. C. Song, G. H. Lee, D.-S. Shim, and K. N. Choi, "Visual distinctiveness detection of pedestrian based on statistically weighting PLSA for intelligent systems," *International Journal of Control, Automation and Systems*, vol. 16, no. 2, pp. 815–822, 2018.

[11] C. Zhang and J. Kim, "Multi-scale pedestrian detection using skip pooling and recurrent convolution," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 1719–1736, 2019.

[12] J. Kim, H. Cho, M. Hwangbo, J. Choi, J. Canny, and Y. P. Kwon, "Deep traffic light detection for self-driving cars from a large-scale dataset," in *Proceedings of the 2018*

21st International Conference on Intelligent Transportation Systems (ITSC), pp. 280–285, Maui, HI, USA, November 2018.

[13] K. Behrendt, L. Novak, and R. Botros, "A deep learning approach to traffic lights: detection, tracking, and classification," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1370–1377, Singapore, May-June 2017.

[14] Y. Lu, J. Lu, S. Zhang, and P. Hall, "Traffic signal detection and classification in street views using an attention model," *Computational Visual Media*, vol. 4, no. 3, pp. 253–266, 2018.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[16] W. Liu et al., "SSD: single shot MULTIBOX detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905, pp. 21–37, Springer, Cham, Switzerland, 2016.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016, pp. 779–788, Las Vegas, NV, USA, June-July 2016.

[18] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[19] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, vol. 2015, pp. 1440–1448, Las Condes, Chile, December 2015.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 2015, pp. 91–99, 2015.

[21] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017, pp. 6517–6525, Honolulu, HI, USA, July 2017.

[22] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 017.

[24] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *Proceedings of the 2nd International Conference on Learning Representations*, Banff, Canada, April 2014.

[25] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification," in *Proceedings of the 23rd international conference on Machine Learning–ICML '06*, pp. 369–376, Pittsburgh, PA, USA, June 2006.

[26] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*vol. 39, no. 11, pp. 2298–2304, 2017.

[27] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence. AAAI 2016*, pp. 3501–3508, Phoenix, Arizona, USA, February 2016.

[28] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016, pp. 2231–2239, Las Vegas, NV, USA, June 2016.

[29] H. Lin, P. Yang, and F. Zhang, "Review of scene text detection and recognition," *Archives of Computational Methods in Engineering*, vol. 27, no. 2, pp. 433–454, 2020.

[30] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): a state-of-the-art review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311–325, 2013.

[31] D. Chai and Y. Zuo, "Extraction, segmentation and recognition of vehicle's license plate numbers," *Advances in Intelligent Systems and Computing*, vol. 887, pp. 724–732, 2019.

[32] J.-K. Chang, S. Ryoo, and H. Lim, "Real-time vehicle tracking mechanism with license plate recognition from road images," *The Journal of Supercomputing*, vol. 65, no. 1, pp. 353–364, 2013.

[33] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, June 2015.

[34] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 850–855, Hong Kong, China, August 2006.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016, pp. 770–778, Las Vegas, NV, USA, June 2016.

[36] F. Zhang and Z. Xu, "A mine image reconstruction method based on residual neural network," *Meitan Xuebao/Journal China Coal Soc.*vol. 44, no. 11, pp. 3614–3624, 2019.

[37] F. Zhang, Z. Xu, W. Chen et al., "An image compression method for video surveillance system in underground mines based on residual networks and discrete wavelet transform," *Electronics*, vol. 812 pages, 2019.

[38] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017, pp. 936–944, Honolulu, HI, USA, July 2017.

[39] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: common objects in context," in *Lecture Notes in Computer Science*, vol. 8693, pp. 740–755, Springer, Cham, Switzerland, 2014.

[40] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: a metric and a loss for bounding box regression," 2019, https://arxiv.org/abs/1902.09630.

[41] S. Hochreiter and J. Urgen Schmidhuber, "Long shortterm memory," *Neural Computation*, vol. 9, no. 8, Article ID 17351780, 1997.

[42] D. P. Kingma, J. L. Ba, and " Adam, "A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, pp. 1–15, San Diego, CA, USA, May 2015.

[43] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *Proceedings of the 5th International Conference on Learning Representations ICLR 2017*, pp. 1–16, Toulon, France, April 2019.

[44] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," pp. 1–10, 2014.

[45] R. Nagy, A. Dicker, and K. Meyer-Wegener, "NEOCR: a configurable dataset for natural image text recognition," *Camera-Based Document Analysis and Recognition*, vol. 7139, pp. 150–163, 2012.

[46] S. Ruder, "An overview of gradient descent optimization," pp. 1–14, 2016.

[47] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[48] M. Everingham, S. M. Ali Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: a retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[49] D. Karatzas, F. Shafait, S. Uchida et al., "ICDAR 2013 robust reading competition," in *Proceedings of the 12th International Conference on Document Analysis and Recognition ICDAR*, pp. 1484–1493, Barcelona, Spain, January 2013.

[50] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proceedings of the 23rd International Conference on Machine Learning ICML 2006*, vol. 2006, pp. 233–240, Pittsburgh, PA, USA, June 2006.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 9453586, 11 pages
https://doi.org/10.1155/2020/9453586

Research Article

Improvement and Application of Generative Adversarial Networks Algorithm Based on Transfer Learning

Fangming Bi,[1,2] Zijian Man,[1,2] Yang Xia,[1,2] Wei Liu,[1,2] Wenjia Yang,[1,2] Xuanyi Fu,[1,2] and Lei Gao[1,2]

[1]*School of Computer Science and Technology, China University of Mining Technology, Xuzhou 221000, China*
[2]*Mine Digitization Engineering Research Center of the Ministry of Education, China University of Mining and Technology, Xuzhou 221116, China*

Correspondence should be addressed to Yang Xia; yxia@cumt.edu.cn

Received 19 April 2020; Revised 28 May 2020; Accepted 3 June 2020; Published 13 July 2020

Guest Editor: Chi-Hua Chen

Generative adversarial networks are currently used to solve various problems and are one of the most popular models. Generator and discriminator are characteristics of continuous game process in training. While improving the quality of generated pictures, it will also make it difficult for the loss function to be stable, and the training speed will be extremely slow compared with other methods. In addition, since the generative adversarial networks directly learns the data distribution of samples, the model will become uncontrollable and the freedom of the model will become too large when the original data distribution is constantly approximated. A new transfer learning training idea for the unsupervised generation model is proposed based on the generation network. The decoder of trained variational autoencoders is used as the network architecture and parameters to generative adversarial network generator. In addition, the standard normal distribution is obtained by sampling and then input into the model to control the degree of freedom of the model. Finally, we evaluated our method on using the MNIST, CIFAR10, and LSUN datasets. The experiment shows that our proposed method can make the loss function converge as quickly as possible and increase the model accuracy.

1. Introduction

GAN is a new image-generating model based on game theory, which innovatively combines the generative model and adversarial model, and proposes a useful training method based on model features to make the output resulting images clearer and sharper than other methods.

Compared with the previous complex image generation methods, the generation of adversarial network does not need to model its original dataset, but only needs to use generators to approach the original data distribution. Its generator and discriminator also do not need complex network structure, and the original deep neural network can achieve better generation effect. Although researchers have made a lot of improvements to the generation of adversarial network, there are still some points that need to be improved based on its own characteristics. For example, the model training speed is slow and the model freedom is too large. So, the purpose of our study is speeding up model training and reducing model freedom.

This is a challenging problem because (1) the generation of adversarial network is a process of continuous game between the generator and discriminator during training. While constantly improving the generation of images, it is difficult for the model to reach equilibrium or even stability due to the violent fluctuation of gradient, and the training speed will be too slow. Traditional GAN training on the CPU takes about 3 hours. (2) The generation of adversarial network requires no modeling in advance. Although the process of generating pictures is simplified, the model is too uncontrollable and the freedom of the model is too large.

To counter the problems above, we use the transfer learning method in the unsupervised deep model and combine with other variational autoencoders of the image

generation model to reduce the resource consumption and time of training generative adversarial network. In addition, based on the particularity of the decoder input of the variational autoencoder, the freedom of the model is further limited by limiting the input data distribution of the generator, which provides an improved method for the future training of other unsupervised generator models. In recent years, based on the great potential of generating adversarial network, the method has been widely improved and applied by researchers. Some studies use GAN in semisupervised learning. Finally, some research studies combine GAN with specific applications and achieve good results.

We combine transfer learning and unsupervised learning to make up for the shortcomings of both sides and use standard normally distributed sampling as parameters to increase input of the model. The specific combination and control methods are as follows:

(1) The decoder of the trained variational autoencoders is used as the network architecture and initial parameters to generative adversarial network generator. The generator is provided with a new starting point for learning and make the loss function fall down as quickly as possible. Experiments show that the method can reduce the training time by about half.

(2) Take standard normally distributed sampling as the input data of the generator, and the problem of too much freedom of the model is improved by limiting the input data of the model.

2. Related Works

The idea of generative adversarial networks (GANs) is derived from game theory's Nash equilibrium [1]. The model is shown in Figure 1. GAN can be seen as a collection of two networks: a generator G, which is responsible for fetching data close to real data, and D, a discriminator, determines whether the sample is a real sample or a false sample, t is sampled from the real data, and n is the data generated by the generator. Two networks are trained by gaming each other, and the loss function is

$$\min_{G}\max_{D} V(D,G) = E_{t\sim p_{data}(t)}[\log D(t)] + E_{n\sim p_{g}(n)}[\log(1 - D(G(n)))]. \tag{1}$$

GAN can be found from conferences and journals in recent years [2–8]and is already a hot topic in artificial intelligence [9–11]. It has many papers related to GANs. The original GAN [1] requires no Markov chain or expanded approximate inference network during training. DCGAN uses CNN to replace the basic nerve structure in GAN and uses global pooling layer to reduce the computation. PROGAN starts to train and gradually generates clear images from blurred images [12]. SAGAN uses the weighted sum of the elements of all positions as the response of that location by self-attention and better handles the dependencies between different regions [13]; LP-WGAN [14] replaces Lipschitz [15] to lp-norm to constrain weight search and effectively close the generated distribution and the true distribution. WGAN-GP [16] uses improved gradient penalty to solve the problem of associating parameters with limits to achieve real Lipschitz constraints. The LSGAN changes the target function of the model, more accurately describes the loss of the model and solves the problems of low picture quality and unstable training process [17]. F-GAN proofs that any divergence is suitable for generating models [18]. UGAN can remove the source category information retained in the generated image, making the source category of the generated image more difficult to track [19]. LSGAN uses Lipschitz regular conditions to further normalize its loss function to the density of real data [20]. MRGAN takes advantage of the unique geometry of real data, especially manifold information [21]. Geometric GAN [22] uses SVM to separate hyperplanes to maximize margin. RGAN [23]considers that the probability that the real data is true should be reduced because this can reduce the gap between true and false data and make the data smoother.

The autoencoder designs a neural network architecture that imposes bottlenecks on the network and forces the original input to compress knowledge representation. The autoencoder is shown in Figure 2. The autoencoder forces the model to retain only the changes of data required by the reconstruction input, but not the redundancy of input. Therefore, the loss function is referred to as

$$L = \min_{\theta} \|x - f(x)\|^2 + \frac{1}{2}\left(\operatorname{tr}\left(\sum(x)\right) + \mu(x)^T\mu(X) - K - \log\det\left(\sum(x)\right)\right). \tag{2}$$

3. Methods

Transfer learning is to transfer the parameters of the model learned and trained to the new model to help the training of the new model. Most of the data or tasks are correlated. Through transfer learning, the model parameters can be shared with the new model in some way to speed up and optimize the learning efficiency of the model without learning from zero as most networks do. Transfer learning mainly includes the concept of data domain and task. A data

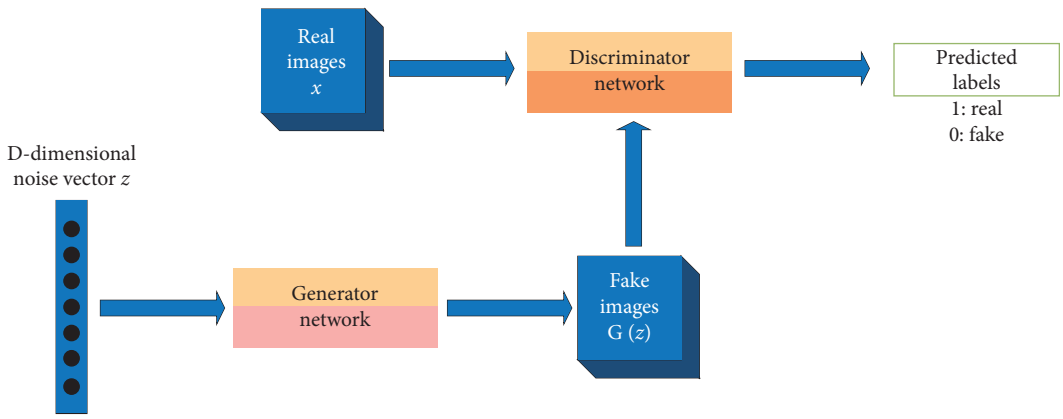


Figure 1: Structural figure of generative adversarial networks.

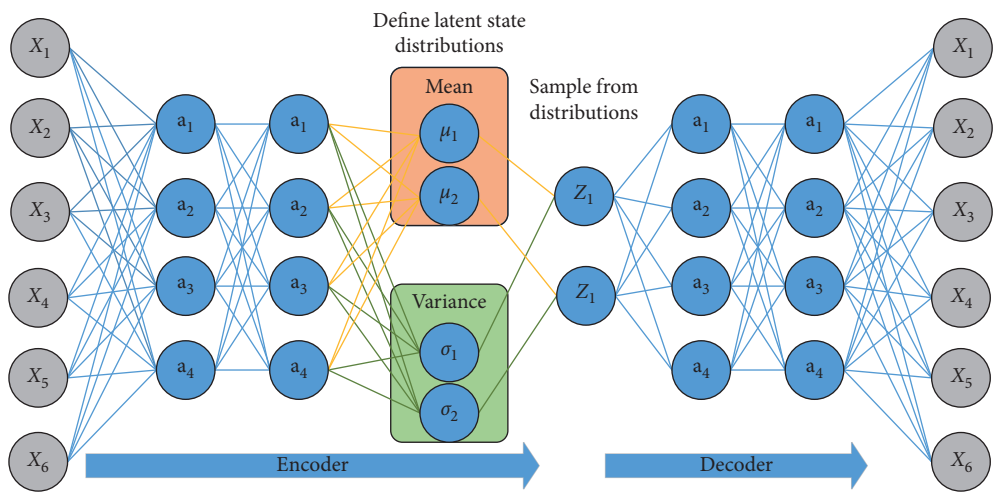


Figure 2: Variational autoencoders include the encoder and decoder.

domain D consists of the feature space X and the probability distribution $P(X)$, and $X = x_1, x_2, \ldots, x_n$. A task T consists of a label space Y and an objective function F. Different models can be transferred according to the correlation between the data domains and tasks of the source model and the target model in transfer learning.

3.1. Transfer Learning Theory Based on CNN. Transfer learning based on convolutional neural network can be divided into two methods.

In the development model, we need to select a relevant predictive modeling problem with rich data first. There is a relationship between the input data of the original task and the target task, the output data, and the concepts learned by adjusting the model parameters with the input data. Then, we need to define an excellent network model for the original task to ensure that the network can enhance the representation of features of original dataset as much as possible. Then, the network model of the original task can be selected as the training starting point of the network model of the new task. Part or all of the original network model can be selected according to the needs of the new task. Finally, according to the training results of the new model and the needs of the target task, the network model can be adjusted appropriately to make it perform better in the new task.

In the pretraining model, we select the source model first. The pretraining model is difference from the development model because the basic model does not need to be trained. Many research institutions have already trained the basic model in the super-large dataset, and the caller only needs to select the network model suitable for the new task. Then, we reuse the model, select the network model of the original task as the training starting point of the network model of the new task, and select part or all of the original network model according to the needs of the new task. Finally, according to the training results of the new model and the needs of the target task, the network model can be adjusted appropriately to make it perform better in the new task.

3.2. Transfer Learning Based on Unsupervised Generation Model. Transfer learning is generally applied to the supervised discriminant model. The source model is

Input: noise data $\{n^{(1)}, \ldots, n^{(m)}\}$ from standard normal distribution $p_{g(n)}$ and $\{t^{(1)}, \ldots, t^{(m)}\}$ from the generated distributed data $p_{\text{data}(t)}$.
Output: The losses of generator and discriminator.
  For generative adversarial network training times do
    For k times do
      Select m small batch samples from standard normal distribution $p_{g(n)}$.
      Select m small batch samples from the generated distributed data $p_{\text{data}(t)}$.
      Optimize discriminator weights by stochastic gradient ascent algorithm: $\nabla\theta_D\,(1/m)\sum_{i=1}^{m}[\log D(t^{(i)}) + \log(1 - D(G(n^{(i)})))]$
    end For
  Select m random samples from standard normal distribution $p_{g(n)}$.
  Optimize generator weights by random gradient descent algorithm: $\nabla\theta_D\,(1/m)\sum_{i=1}^{m}\log(1 - D(G(n^{(i)})))$
end For

Algorithm 1: Improved original generative adversarial network Algorithm. D and G are the two parts that make up the GANs. Really sampled data t and the noise data n. $G(n)$ is the sample generated by G that is closest to real data. $D(t)$ is the discriminator of really sampled data, and $D(G(n))$ is the discriminator of fake data from the generator.

transferred to the target model after training through similar datasets, and a new training starting point is given to the target model to accelerate the training speed of the target model. By combining generative models with transfer learning, a new transfer method is presented which can be applied to the unsupervised generation model of deep learning. In this paper, two unsupervised generation models of generation adversarial network and variational autoencoder and their improvement are studied, and MNIST and LSUN datasets are selected as input data. The variational autoencoder is selected as the source model of transfer learning, and the original generative adversarial network and WGAN-GP are selected as the target model. This experiment adopts a new training method of transfer learning based on the unsupervised generation model:

(1) Select the source model: the variational autoencoder is chosen as the source model of training
(2) Reuse model: the model parameters of the decoder of the variational autoencoder are saved
(3) Adjust model: the generative adversarial network is adjusted so that the input of the generative model is obtained from the standard positive attitude distribution, and then the iterative training is carried out according to the generative adversarial network algorithm

The improved original generation adversarial network adopts the small-batch stochastic gradient algorithm. The training times of the discriminator are k, which is a hyperparameter. The dataset is input into the encoder of the variational autocoder so that the encoder learns mean and variance. The potential variable n is selected from the standard normal distribution and input into the decoder to train the variational autocoder so that its loss function is stable or even minimal. The network structure and network parameters of the decoder of the variational autocoder after training are transferred to the generative adversarial network as its initial generator for the following training. The improved original generative adversarial network algorithm is described in Algorithm 1.

4. Experiments

In this experiment, the MNIST, CIFAR10, and LSUN datasets are used to compare different models. The MNIST is a computer vision dataset, which contains 70,000 grayscale images of handwritten digits, each of which contains $28 * 28$ pixels, and each image has a corresponding label that is the corresponding number of images. The CIFAR10 has a total of 60,000 color images, and the image pixels are $32 * 32$, which are divided into 10 categories with 6,000 images in each category. The LSUN mainly includes 10 scene categories and 20 object categories, and each category has about one million pictures. It is a scene understanding image dataset, mainly including bedroom, house, living room, classroom, and other scene images. We improve the generative adversarial network, deep convolutional network, and WGAN-GP, respectively, in three datasets. The details are shown in Table 1. The original generative adversarial network trains the MNIST dataset, and the improved original generative adversarial network adopts the same network structure. The leaky ReLU activation function is used in the original network activation function. Deep convolution generative adversarial network is to change the structure of the original generative adversarial network. The improved deep convolution generative adversarial network adopts the same network structure. In the discriminator of deep convolution generative adversarial network, step convolution is used to replace pooling layer for down sampling, and in the generator, transposed convolution is used to replace upsampling. Removing the full connection layer and replacing the full connection layer with global pooling, the improved WGAN-GP and WGAN-GP adopt the same network structure as deep convolution generative adversarial network, while the WGAN-GP improves the loss function of network to avoid the model collapse.

4.1. Improved Original Generative Adversarial Network. In this experiment, the low-pixel MNIST dataset is used for training, and the network structure of the generator and discriminator is relatively simple. In order to make a comparison with the original generative adversarial

Table 1: The experimental comparison table.

Dataset	Algorithm	Instruction
MNIST	Original generative adversarial network Improved original generative adversarial network	The experimental results of the original network and the improved network are shown, which show the effectiveness of improved network
CIFAR10	Deep convolution generative adversarial network Improved deep convolution generative adversarial network	
LSUN	WGAN-GP Improved WGAN-GP	

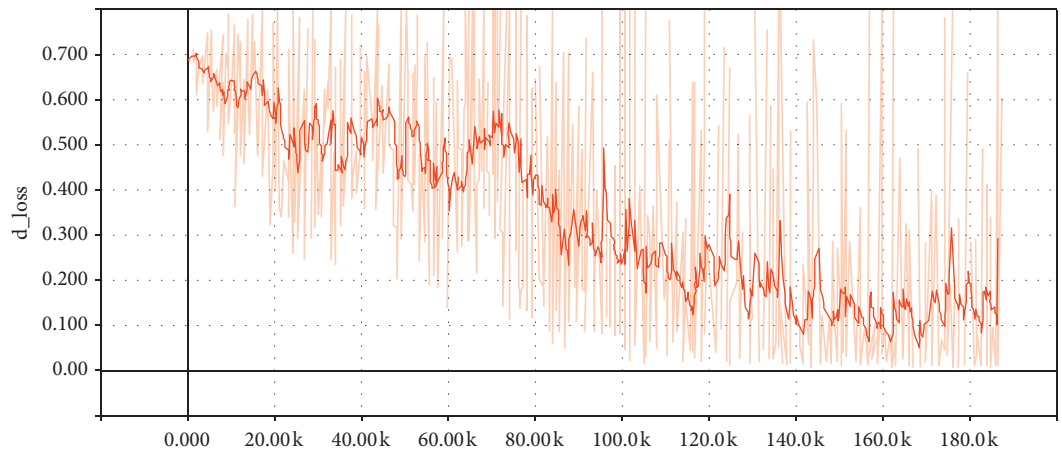


Figure 3: The discriminator loss function of generative adversarial networks.

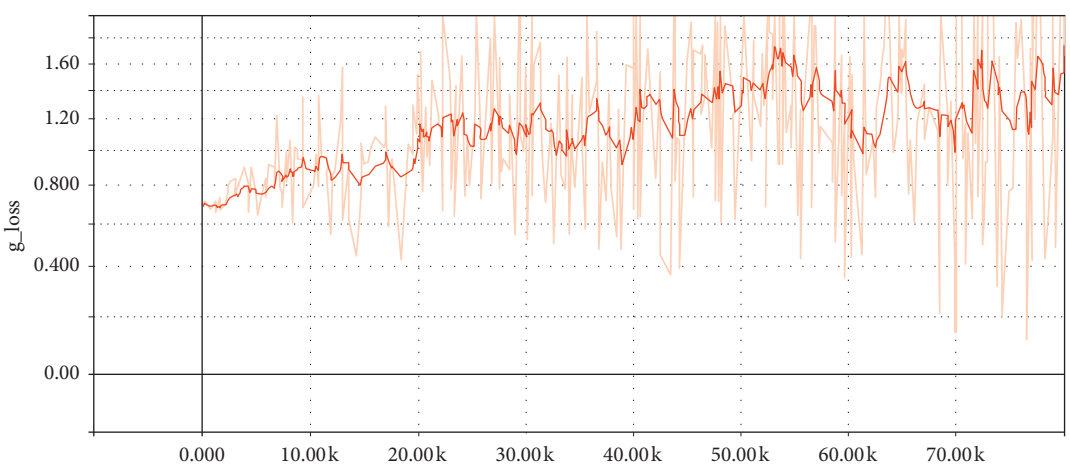


Figure 4: The generator loss function of generative adversarial networks.

network, this experiment uses the MNIST dataset to train the two algorithms, respectively. The MNIST dataset generate the training result graph of the adversarial network in the original phase in Figures 3 and 4. The MNIST dataset in the generation of improved counter network training loss in Figures 5 and 6. Figures 7(a) and 7(b) show the generated images, respectively. In the experimental figure, *d_loss* is the loss value of the discriminator, *g_loss* is the loss value of the generator, and the x-coordinate is the number of iterations during training. As the training time increases, the loss of the network gradually decreases and tends to be stable when the training times of the original generative adversarial network reach to 120 k. And generative adversarial network in training reach to 40k, network convergence, generator, and discriminant loss function tends to be stable; therefore, when using a transfer generative adversarial network-based algorithm while the MNIST dataset for network training and greatly improves the convergence speed.

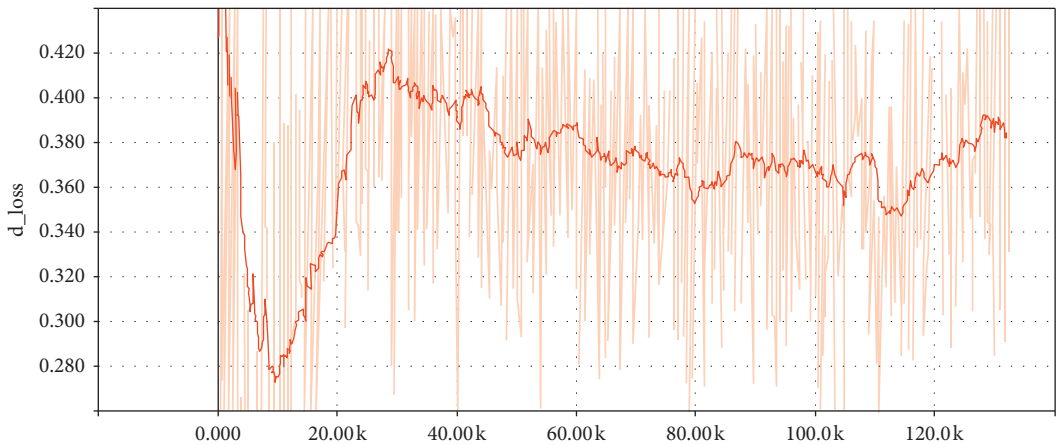


Figure 5: The discriminator loss function of improved generative adversarial networks.

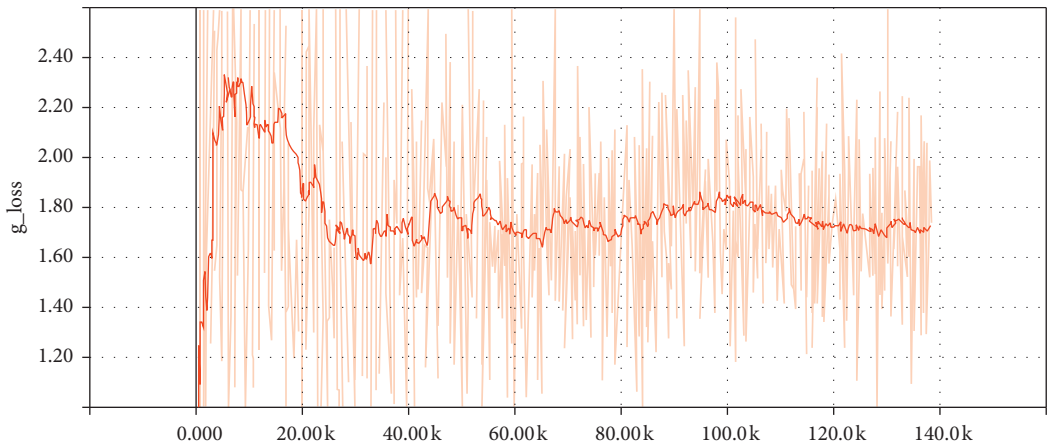
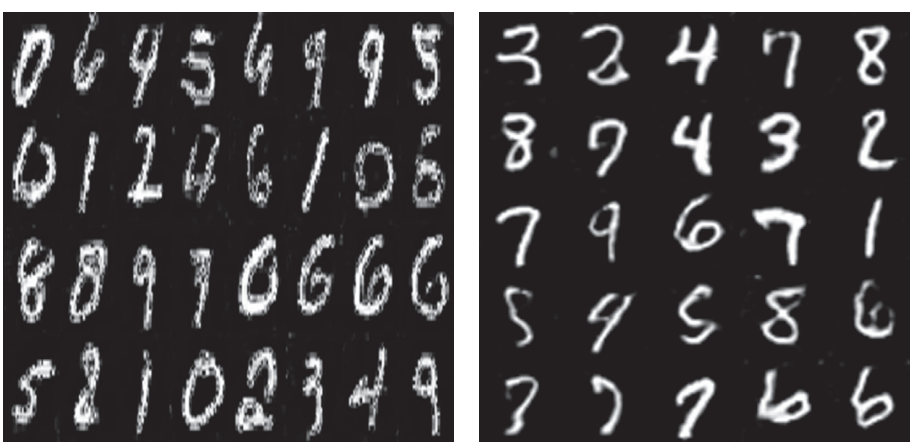


Figure 6: The generator loss function of improved generative adversarial networks.



(a) (b)

Figure 7: (a) Generative adversarial network-generated pictures. (b) Improved generative adversarial network-generated pictures.

4.2. Improved Deep Convolution Generative Adversarial Network. In this experiment, we use the CIFAR10 for training and deep convolution which is adopted by the generator and discriminator to generative adversarial network. For the purpose of making a comparison with the adversarial network generated by the original deep

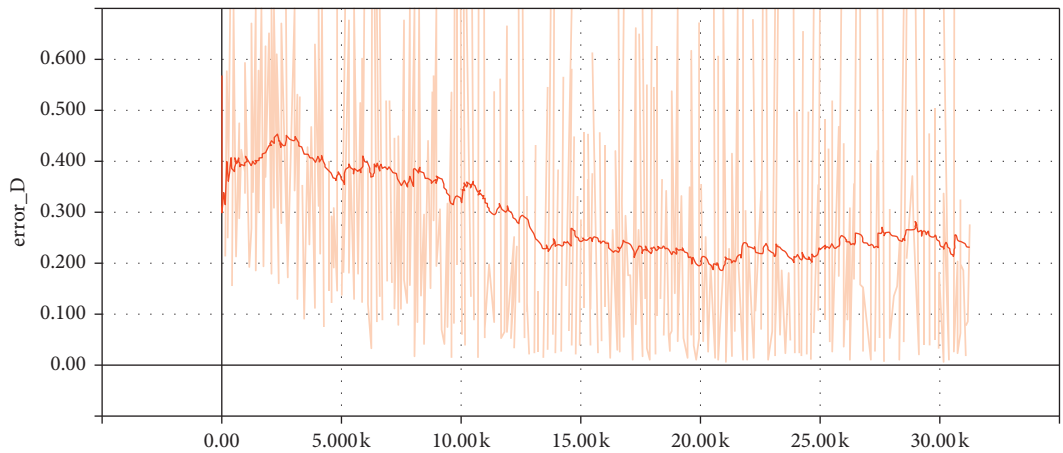


Figure 8: The discriminator loss function of deep convolutional generative adversarial networks.

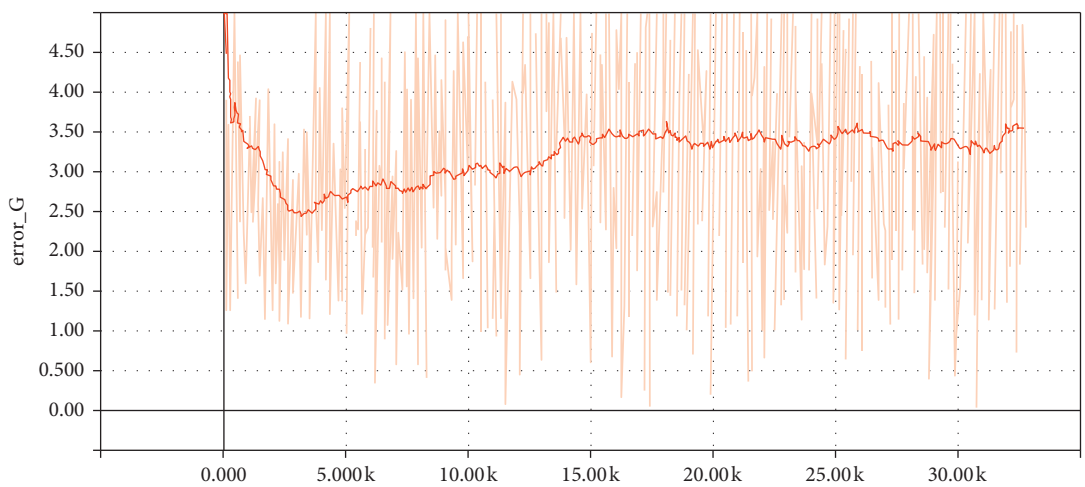


Figure 9: The generator loss function of deep convolutional generative adversarial networks.

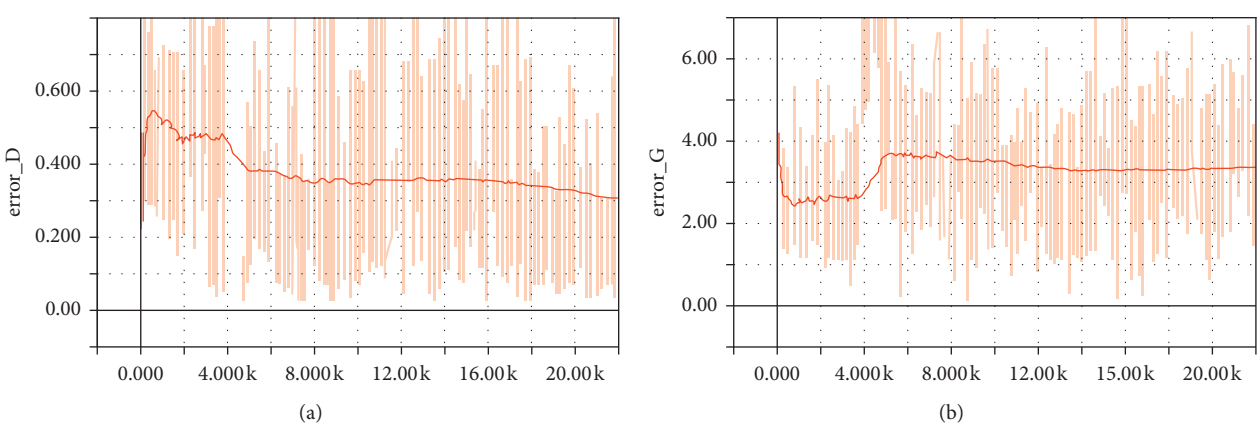


Figure 10: (a) The discriminator loss function of improved deep convolutional generative adversarial networks. (b) The generator loss function of improved deep convolutional generative adversarial networks.

convolution, this experiment uses the CIFAR10 dataset to train the two algorithms, respectively. The CIFAR10 dataset generates training result graph of countermeasure network in the original deep convolution in Figures 8 and 9. The CIFAR10 dataset generates the training result graph of adversarial network in the improved deep convolution in Figures 10(a) and 10(b). Generated images are shown in Figures 11 and 12. In the experimental figure, *error_D* is the

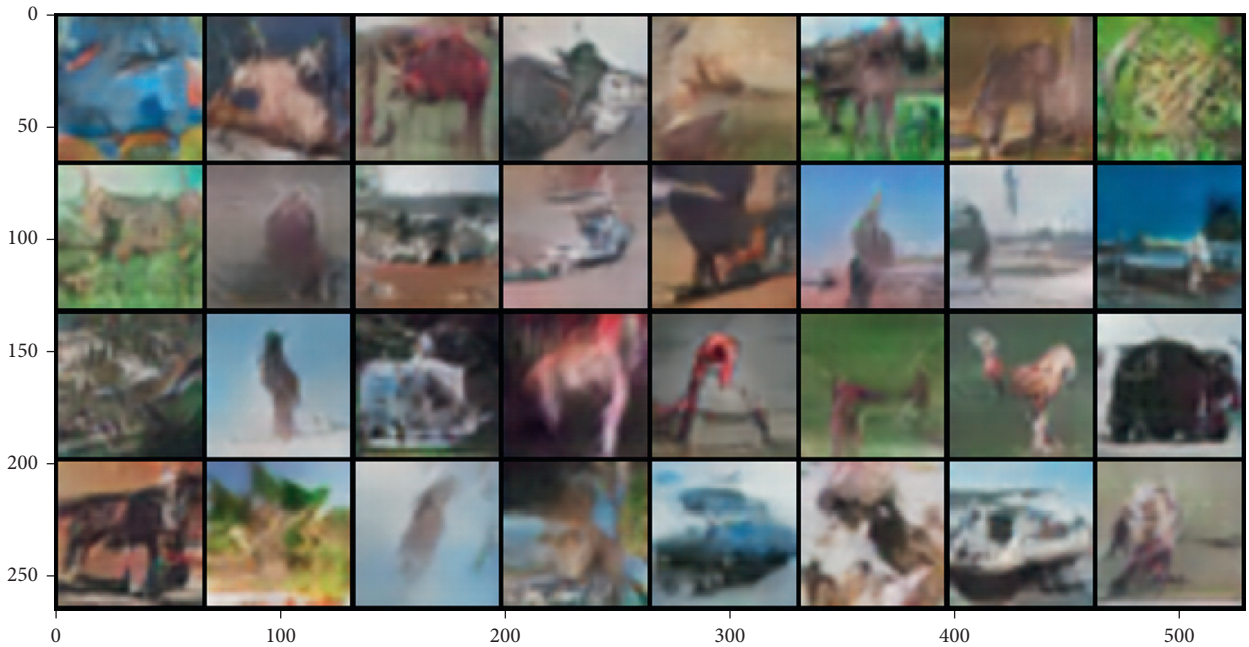


Figure 11: Deep convolutional generative adversarial network-generated pictures.

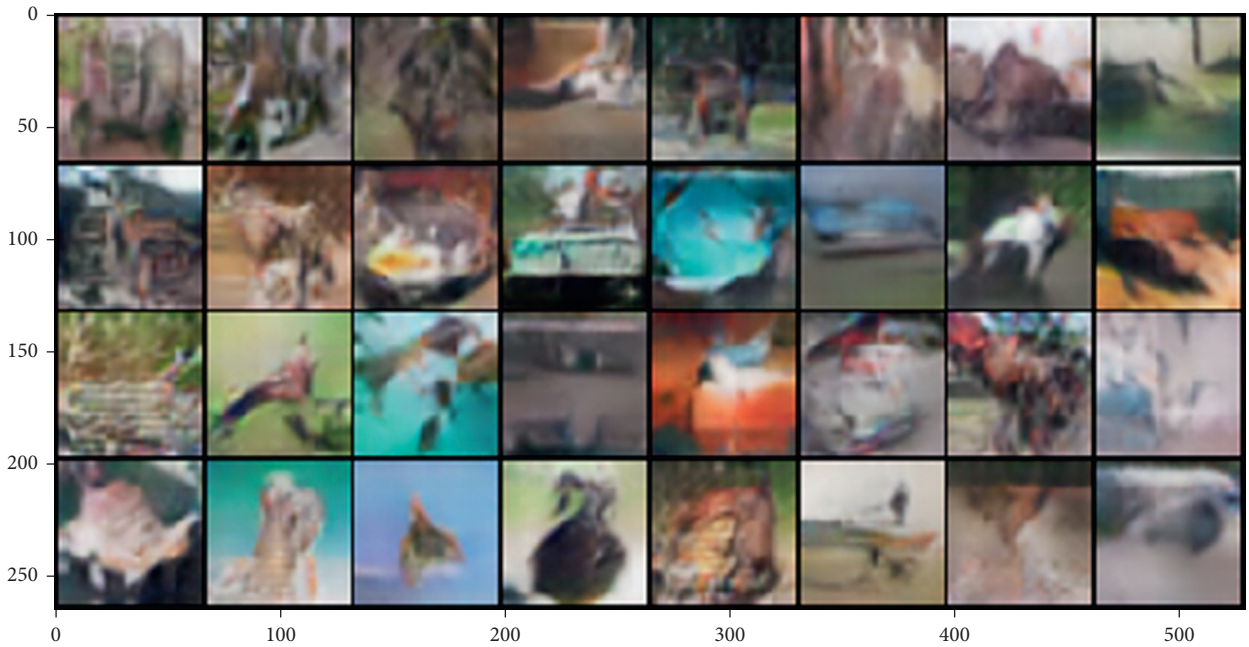


Figure 12: Improved deep convolutional generative adversarial network-generated pictures.

loss value of the discriminator, *error_G* is the loss value of the generator, and the x-coordinate is the number of iterations during training. When the training times of the original deep convolution generation adversarial network reaches 15 k, the network reaches convergence and the loss function of the generator and discriminator tends to be stable and improves the deep convolution generative adversarial network in training reach to 8k, network convergence, generator, and discriminant loss function tends to be stable. Therefore, when deep convolution generative adversarial network based on transfer learning is used for training in CIFAR10, the convergence speed is improved. In addition, from the result graph of generated images, the quality of generated images is also improved to some extent.

4.3. Improved WGAN-GP. In this experiment, the LSUN dataset is used for training, and the generator and discriminator used deep convolution to generate the adversarial network. In order to make a comparison with the original

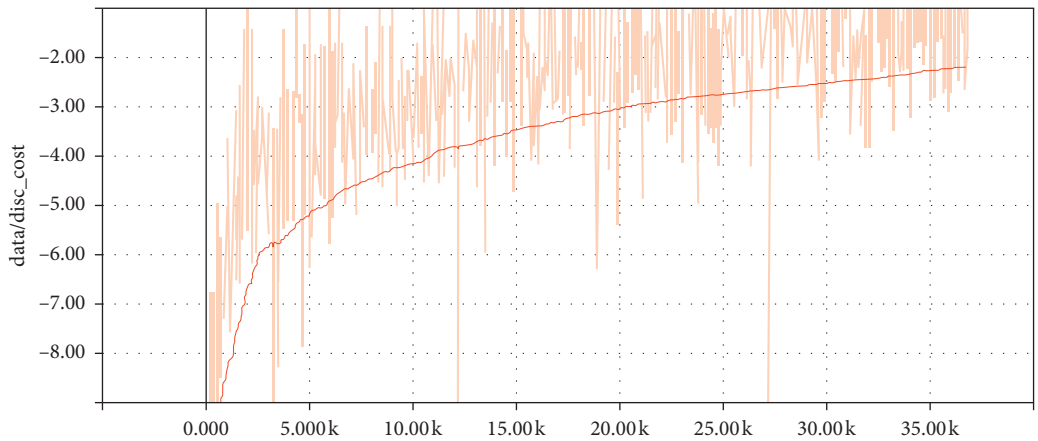


Figure 13: WGAN-GP discriminator loss function.

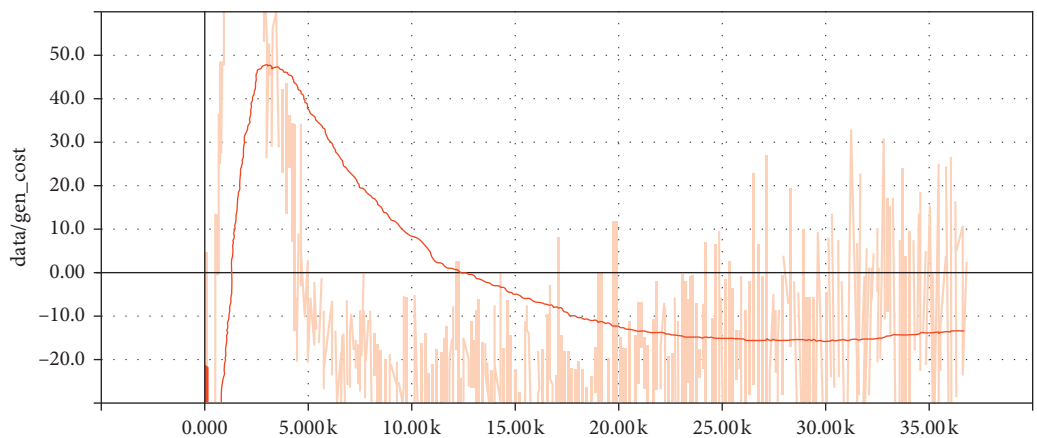


Figure 14: WGAN-GP generator loss function.

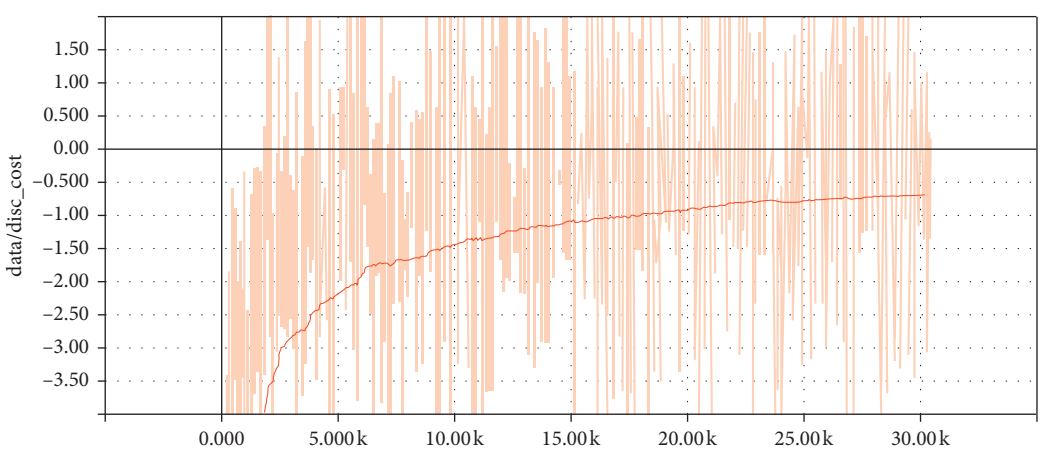


Figure 15: Improved WGAN-GP discriminator loss function figure.

WGAN-GP. The LSUN dataset is used for training in the two algorithms in this experiment. The LSUN dataset in the original WGAN-GP training result graphs is shown in Figures 13 and 14. The LSUN dataset in the improved WGAN-GP training result graphs are shown in Figures 15 and 16. Generated images are shown in Figures 17 and 18, respectively. In the experimental figure, *data/disc_cost* is the loss value of the discriminator, *data/gen_cost* is the loss value of the generator, and the x-coordinate is the number of iterations during training. In the loss function graph of the generator and discriminator, when the training frequency of original deep WGAN-GP reaches 25k, the network reaches

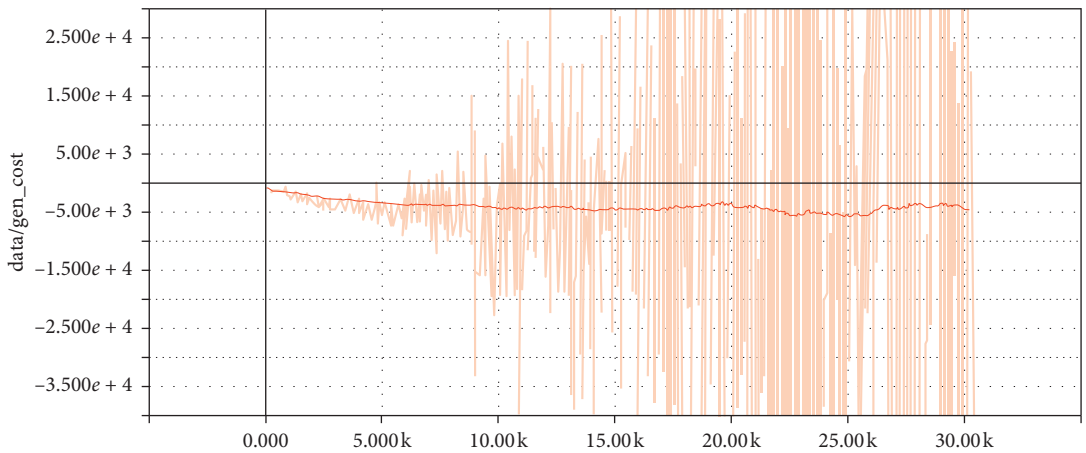


Figure 16: Improved WGAN-GP generator loss function.

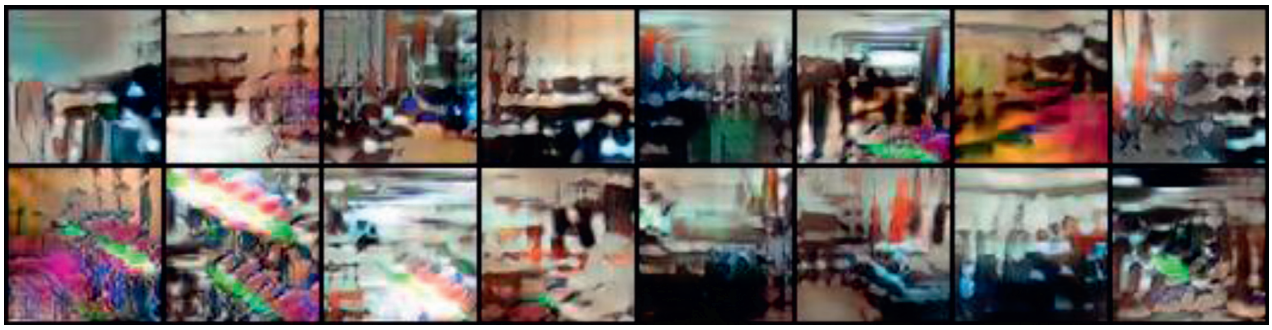


Figure 17: WGAN-GP generated pictures.

Figure 18: Improved WGAN-GP generated pictures.

convergence and the loss function of the generator and discriminator tends to be stable. And improved WGAN-GP in training reaches to 20 k, network convergence, generator, and the discriminant loss function tends to be stable; therefore, when using WGAN-GP algorithm based on migration study in LSUN dataset for network training, improve the convergence speed, the other by the result of the generated image, and generated images are also better than the previous images, and images become more acute.

5. Conclusions

In all unsupervised generative models, the generation of adversarial network has the advantages of clearer and sharper image generation, and does not need to use Markov chain to sample the input dataset repeatedly, and the model is more concise than other unsupervised network models. However, the problems of long training time and too much freedom of the model have not been improved. So, we put forward the transfer method between unsupervised generation models. First, we transfer the decoder of the variational autocoder to the generator that generates the adversarial network and the convergence speed of the adversarial network is greatly accelerated. Second, our model restricts the generation of the input of the antagonistic network and uses the standard normal distribution, which further limits the freedom of the model and makes it more instructive and closer to the real picture. Experiments show that generated

pictures are more instructive and closer to the real pictures. The image resolution is still the focus of research, so we will design experiment to building a new network that makes the generated images more realistic and speeds up training in our future work.

Data Availability

The MNIST, CIFAR10, and LSUN data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the Open Program of Shanghai Key Lab of Intelligent Information Processing under Grant no. IIPL-2019-10.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial networks," 2014, https://arxiv.org/abs/1406.2661.

[2] O. Goudet, D. Kalainathan, P. Caillou et al., "Learning functional causal models with generative neural networks," *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pp. 39–80, Springer Link, New York, NY, USA, 2018.

[3] Y.-Y. Zhang, C.-M. Shen, H. Feng, P. T. Fletcher, and G.-X. Zhang, "Generative adversarial networks with joint distribution moment matching," *Journal of the Operations Research Society of China*, vol. 7, no. 4, pp. 579–597, 2019.

[4] S. Song, W. Zhang, J. Liu, and T. Mei, "Unsupervised person image generation with semantic parsing transformation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2357–2366, IEEE, Long Beach, CA, USA, June 2019.

[5] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 2642–2651, Sydney, Australia, July 2017.

[6] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6721–6729, IEEE, Honolulu, HI, USA, July 2017.

[7] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proceedings of the European Conference on Computer Vision*, pp. 597–613, Springer Link, New York, NY, USA, September 2016.

[8] C. Lassner, G. Pons-Moll, and P. V. Gehler, "A generative model of people in clothing," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 853–862, IEEE, Venice, Italy, October 2017.

[9] H. Gan, Z. Luo, M. Meng, Y. Ma, and Q. She, "A risk degree-based safe semi-supervised learning algorithm," *International Journal of Machine Learning & Cybernetics*, vol. 7, no. 1, pp. 85–94, 2016.

[10] W. Fedus, I. Goodfellow, and A. M. Dai, "MaskGAN: better text generation via filling," 2018, https://arxiv.org/abs/1801.07736.

[11] N. Jetchev, U. Bergmann, and R. Vollgraf, "Texture synthesis with spatial generative adversarial networks," 2016, https://arxiv.org/abs/1611.08207.

[12] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8789–8797, IEEE, Salt Lake City, UT, USA, June 2018.

[13] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, https://arxiv.org/abs/1805.08318.

[14] C. Zhou, J. Zhang, and J. Liu, "Lp-WGAN: using Lp-norm normalization to stabilize Wasserstein generative adversarial networks," *Knowledge-Based Systems*, vol. 161, pp. 415–424, 2018.

[15] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[16] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang, "Improving the improved training of wasserstein gans: a consistency term and its dual effect," 2018, https://arxiv.org/abs/1803.01541.

[17] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, IEEE, Venice, Italy, October 2017.

[18] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: training generative neural samplers using variational divergence minimization," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 271–279, Barcelona, Spain, 2016.

[19] Y. Wang, B. Yu, L. Wang et al., "3D conditional generative adversarial networks for high-quality PET image estimation at low dose," *NeuroImage*, vol. 174, pp. 550–562, 2018.

[20] C. Wu, L. Li, Z. Yang, P. Yan, and J. Jiao, "Image-to-Image local feature translation using double adversarial networks based on CycleGAN," in *Proceedings of theInternational Conference in Communications, Signal Processing, and Systems*, pp. 907–915, Springer Link, New York, NY, USA, August 2019.

[21] Y. Li, N. Xiao, and W. Ouyang, "Improved generative adversarial networks with reconstruction loss," *Neurocomputing*, vol. 323, pp. 363–372, 2019.

[22] J. H. Lim and J. C. Ye, "Geometric GAN," 2017, https://arxiv.org/abs/1705.02894.

[23] A. Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard GAN," 2018, https://arxiv.org/abs/1807.00734.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 4315850, 13 pages
https://doi.org/10.1155/2020/4315850

Research Article

An Optimization Technique of the 3D Indoor Map Data Based on an Improved Octree Structure

Xiaomin Yu,[1,2] **Huiqiang Wang**,[1] **Hongwu Lv**,[1] **and Junqiang Fu**[1]

[1]*College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang, China*
[2]*College of Computer and Control Engineering, Qiqihar University, Qiqihar, Heilongjiang, China*

Correspondence should be addressed to Huiqiang Wang; wanghuiqiang@hrbeu.edu.cn

Received 15 April 2020; Revised 2 June 2020; Accepted 10 June 2020; Published 10 July 2020

Guest Editor: Chi-Hua Chen

The construction and retrieval of indoor maps are important for indoor positioning and navigation. It is necessary to ensure a good user experience while meeting real-time requirements. Unlike outdoor maps, indoor space is limited, and the relationship between indoor objects is complex which would result in an uneven indoor data distribution and close relationship between the data. A data storage model based on the octree scene segmentation structure was proposed in this paper initially. The traditional octree structure data storage model has been improved so that the data could be backtracked. The proposed method will solve the problem of partition lines within the range of the object data and improve the overall storage efficiency. Moreover, a data retrieval algorithm based on octree storage structure was proposed. The algorithm adopts the idea of "searching for a point, points around the searched point are within the searching range." Combined with the octree neighbor retrieval methods, the closure constraints are added. Experimental results show that using the improved octree storage structure, the retrieval cost is 1/8 of R-tree. However, by using the neighbor retrieval, it improved the search efficiency by about 27% on average. After adding the closure constraint, the retrieval efficiency increases by 25% on average.

1. Introduction

Nowadays, people have higher requirements for using the electronic map due to the popularization of electronic map application and the rapid development of related technologies. The traditional two-dimensional map cannot meet the latest requirements in terms of visual effects. Surveys showed that people are spending more time in indoor environment nowadays, including shopping malls, exhibition halls, libraries, cruise ships, and many other large-scale indoor environments [1]. The existence of three-dimensional indoor map is of practical significance [2]. Data storage efficiency, retrieval consumption, time delay, and other parameters will directly affect the application level of the maps [3]. Consequently, the efficient management of the indoor map data has become a major challenge.

At present, all walks of life, including the fields of map, are faced with the problem that huge data is difficult to store and complex to query [4]. Since the data sets of a map scene are very large, they are not able to be stored directly by the indexes. Therefore, a scene segmentation method is needed as a secondary index to reduce the data storage unit and achieve the lightweight storage. Generally speaking, due to the different environment, maps can be divided into outdoor maps and indoor maps. The regional segmentation technology is preferable as the outdoor scene is large and uneven, and the map data is stored as sheet. Compared with the outdoor map, the indoor map includes numerous objects, and the volume of the objects is small. The situation of uneven distribution often exists. If the sheet framing is used, a large number of objects may be distributed in a sheet. This extreme situation will reduce the regional segmental effect. In order to solve this problem, it is necessary to continue the segmentation, which will increase the cost of scene segmentation. The sheet framing cannot meet the vertical requirement between different floors and cannot preserve the three-dimensional nature of the data.

The technology of the storage of outdoor map data has developed rapidly, while the indoor map data storage is still in the initial stage [5] of development. For the indoor data storage, the scene can be divided by tree structure, and the octree is one of the high-efficiency scene segmentation methods, which can realize the rapid segmentation of 3D map scene and retain the characteristics of map data [6]. However, at present, this method is mainly applied in the field of game scene and 3D image processing, and there are few applications for indoor map. Moreover, indoor map data is updated more frequently, and there are room partitions between the data. Therefore, the traditional scene segmentation methods are not applicable for the indoor map data [7]. One of the key problems is to design a set of storage structure suitable for indoor map features based on fast scene segmentation method.

The efficiency of the octree structure in the scene segmentation is very high. However, its performance in data retrieval is not ideal. After scene segmentation using octree structure, map data will be stored in corresponding nodes. We need to traverse every required data tree, so the efficiency of querying and updating indoor map data is not high. This leads to another key problem, which is improving the efficiency of data retrieval and ensuring high storage efficiency.

In this paper, a set of indoor map data storage model based on the scene segmentation is designed, which can fulfill the scene segmentation, code design, data expression, and storage structure design. And a fast data retrieval method based on the data storing models is also proposed. These results provide an efficient data storage and retrieval scheme.

2. Related Works

At present, map data processing software (such as ArcGIS) or domestic emerging software (including SuperMap) mostly adopt the scene segmentation for data unit division and form a map data index structure. The ArcGIS SDE of Esri supports fixed grid scene segmentation structure, while MapGIS and SuperMap support quad-tree scene segmentation structure. Furthermore, ArcView and MapInfo support R-tree scene segmentation structure, while Oracle Spatial supports both R-tree and quad-tree scene segmentation structure [8].

Fixed grid is a kind of scene segmentation structure which is widely used in two-dimensional space environment. Its specific principle is to use a preset size mesh to segment the scene and form a rectangular pixel of equal size. Each rectangular cell is uniquely annotated and recorded according to the row number and the column number. The contents stored in the rectangle cell area are related to the information and properties of the data. The data in the whole scene find their respective storage locations through uniquely marked rectangular area [9].

And when it is necessary to retrieve the data from the scene, the location of the cell can be located by querying the row number and the column number. Since there may be multiple data in each cell, it is also necessary to compare the data attribute values to determine the retrieved data. According to the research conducted by Qiao [10], data retrieval with fixed grid method can be divided into accurate retrieval and range retrieval. Accurate retrieval can complete the search of row number and column number to uniquely determine the cell location, but it still needs range retrieval to continue to retrieve all units that intersect the range retrieved in the first step. One of the studies conducted by Yang [11] pointed out that due to the variety of map data, when the data attributes were particularly complex, it would lead to the redundancy of the index records and reduce the efficiency of data retrieval. This method is similar to the rectangle framing and longitude latitude framing in the map, and it is more suitable for two-dimensional space.

Originally as an index method, R-tree structure was proposed by Guttman in 1984. This structure is applicable to both two-dimensional environment and three-dimensional environment, and it is an extension of B-tree [12]. Many researches have been carried out on the deformation of R-tree, and specific needs have been met [13]. These studies often expand R-tree node and store some properties similar to topological relationship. However, this method will double the stored information which increases the burden to the storage and cause the problem of data redundancy.

The study carried out by Wang et al. [14] proposed a method combining R-tree and the octree to manage large-scale three-dimensional scene. Its advantage is to implement dynamic, efficient, and real-time management of the change data during the interaction. It makes the rendering picture smoother, resulting in a significant reduction in rendering time. However, due to fusing two types of complex structures, it will consume more time to construct the data structure. The study carried out by Namdari et al. [15] proposed an efficient parallelization task execution method based on R-tree. By designing the storage structure, the problem of limited parallelization effect of R-tree on GPU was solved, and a fast search method of R-tree was proposed. R-tree is one of the commonly used storage methods at present, but the objects in the indoor scene are placed with a large density and uneven distribution, so this method is not much applicable [16].

As a scene segmentation method of tree structure, the quad-tree is mainly applied to two-dimensional scene. It uses the fixed grid structure to divide each space into four rectangular areas. It can achieve more efficient space segmentation compared with the grid structure. Compared with R tree, quad-tree structure is relatively simple, so its construction time is shorter. Due to the regularity of each recursion, each quad-tree node can be encoded. The data retrieval can be completed by encoding. This structure is more suitable for two-dimensional space scenes similar to the fixed grid structure [17].

The octree structure was first proposed by Meagher in 1982 [18]. It is an extension of the quad-tree in three-dimensional environment [19], and it has the characteristic of flexibility and extensibility [20]. It is a popular method in the field of 3D image processing and spatial data index [21] and has a better effect than R-tree [22]. The octree has been applied in the storage of 3D objects data such as iron and

steel industry, etc. [23]. Riegler et al. [24] and Tatarchenko et al. [25] proposed a fast scene reconstruction method by taking advantage of the regularity of the octree segmentation. Deng et al. [26] proposed a method to add an update memo in the octree, which improves the update efficiency and significantly reduces the update cost of 3D objects.

These methods can initially realize the basic storage of 3D spatial data, but due to the large density and uneven distribution of indoor map data, these storage processes have many disadvantages in some aspects.

When the map scene is segmented by the octree, the objects' data is stored in the nodes. In this way, when retrieving objects, it needs to go through the tree. So, it is very tedious for the data with large retrieval requirements. Many scholars have studied the retrieval method, and the current aspect is the retrieval of neighbor nodes. The neighbor query is divided into four classes [27]: (i) all objects are also the query points. This is known as an all-kNN query (akNN); (ii) only selected objects are the query points. This is known as a Group Nearest Neighbor (GNN); (iii) there is a variant of GNN, which is called Group Nearest Group (GNG); and (iv) only object locations within a given rectangle are the query points. This is known as a Range-kNN query. Optimization techniques used include machine learning [28–30], deep learning [31, 32], and optimization technologies for transportation [33–35]. The study carried out by Pan et al. [31] proposed a fine-grained classification model named RMA (ResNet-Multiscale-Attention) based on deep learning to analyze the subtle and local differences among navigation mark types for the recognition of navigation marks. The study carried out by Pan et al. [32] proposed a CNN-GRU model combined with the network structures of gated recurrent unit (GRU) and convolutional neural network (CNN). The GRU part learned the changing trend of water level, and the CNN part learned the spatial correlation among water level data observed from adjacent water stations.

The study carried out by Cho et al. [36] proposed a neighbor retrieval method, which took advantage of the regularity of the octree segmentation and could find out the 26 neighboring nodes by calculating the offset in three directions of x, y, and z. However, the neighbors search was done after the tree is built again. This approach is simple and does not add any time complexity. Furthermore, Namdari et al. improved on this basis [15] and proposed an improved neighbor retrieval method, which stored the neighbors of all leaf nodes in the process of building the tree. It has the advantage of conventional octree partitioning. Storing neighbors when building a tree will not cause damage to the complexity of building tree. They proved that the sum of time for retrieval, storing neighbors, and building tree is not much different from the time it takes to create the tree alone.

Although the neighbor node search method is feasible, it does not need all 26 neighbor nodes for the indoor scene of closed room. There is still some unnecessary consumption, and the efficiency needs to be further improved. In this study, we proposed an octree-based data storage model with backtracking and an octree neighbor retrieval algorithm with closed constraints.

3. Materials and Methods

3.1. Map Scene Segmentation. The octree storage model usually consists of three parts: scene segmentation, node coding, and data representation. The basic principle to fulfill the function is using the octree to segment 3D scene regularly, using octree code to encode the route of each node [37]. The cube boxes are further used to express the objects approximately, and the codes are stored in the corresponding nodes according to the size and location of the boxes [38–40].

3.1.1. Octree Scene Segmentation Technology. The octree is a regular segmentation data structure. It approximately takes the overall scene of a three-dimensional indoor map as a cube and takes it as the root node of the tree. Moreover, it divides the scene into 8 or 0 pieces at a time during scene segmentation. Its schematic diagram is shown in Figure 1, and its definition is as follows:

$$\text{octree} = \begin{cases} v_i = \frac{1}{8} v_{i-1},\ d_i = \frac{1}{2} d_{i-1}, h_i = h_{i-1} + 1, & n > 2, \\ \text{end of separation}, & n \leq 2 \text{ or } d_i = d_{\min}, \end{cases} \tag{1}$$

where v_i, d_i, and h_i are the volume, side length, and depth of the cube obtained by the ith division, respectively, n is the number of objects contained in the cube, and $d_{\min}$ is the minimum side length that the cube can be separated into.

There are three types of nodes in an octree. In order to distinguish them, three colors are used in this paper. The white nodes indicate that no data has been stored, whereas gray nodes indicate that more than two objects' data have been stored and black nodes indicate that one- or two-object data have been stored. White and black nodes can be regarded as leaf nodes, and gray nodes need to be further segmented. This segmentation is a recursive process, whose termination condition is that all nodes are leaf nodes or segmentation reaches the specified minimum segmentation area. Figure 1 shows a twice-segmented tree structure, in which (a) is a schematic diagram of three-dimensional cube, whereas white circles represent the objects, and (b) is a schematic diagram of two-dimensional tree structure.

After a three-dimensional map is recursively segmented by octree algorithm, many subregions will be formed, which are distributed in all the corners of the three-dimensional space.

3.1.2. Octree Node Addressing Encoding. In order to make the data management organized in the octree data storage models, it is necessary to make clear position of each data corresponding to the tree. In this paper, the position of each subregion is marked by encoding. The node addressing code uses digits to mark the 8 child-nodes which were extracted from each node, so that the 3D data can be represented by one-dimensional digits. Among which the number of encoded bits is the same as the number of layers, and each encodes unique marks of the node region while recording

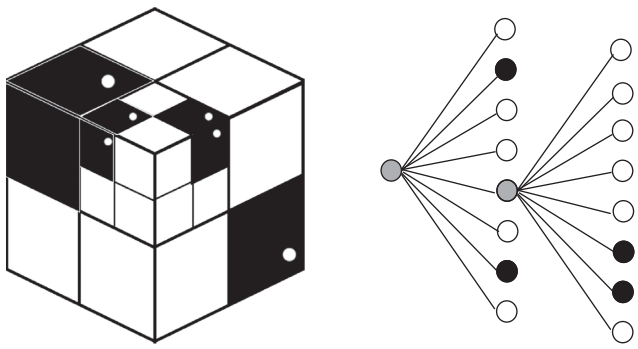


Figure 1: The octree scene segmentation schematic diagram.

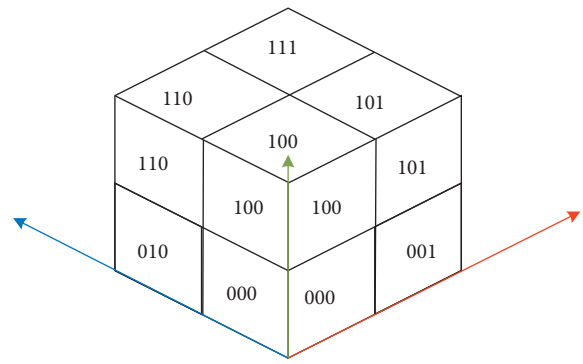


Figure 2: The octree nodes addressing encoding.

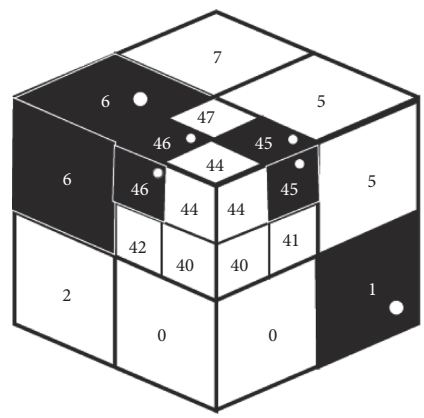


Figure 3: The decimal coded representation of the octree nodes.

the path of this node to the root node. The method of "push back" is used in encoding. The first segmented code is taken as the first digit, and the result of each recursive segmenting is added to the last several digits in turn. The segmented 8 subregions can be distinguished by the deviation degree from the three direction axes, and this method can mark the spatial information in the 8 directions with only 3 binary bits. The following formula can be used to calculate the node addressing encoding, where p is the current number of layers and n is the number of layers to be segmented:

$$\text{coding} = (x_p y_p z_p), (x_{p+1} y_{p+1} z_{p+1}), \ldots, (x_{p+n} y_{p+n} z_{p+n}). \tag{2}$$

The first bit of the three-bit binary code indicates the deviation degree from the x-axis, 0 indicates the subregion has no deviation in the x-axis direction, and 1 indicates the subregion has deviation in the x-axis direction. Similarly, the second bit and the third bit represent the deviation degree of the subregion in the y-axis and z-axis direction, respectively, which is exhibited in Figure 2.

The maximum number of layers for node segmentation is determined by the system word length. m is used to indicate the maximum number of layers that the current system can be divided into, and *Bits* represents the number of bits in the system word length:

$$m = \frac{\text{bits}}{3}. \tag{3}$$

A 64-bit system can be divided up into 21 layers. Assuming that the required minimum partition area is $1\,\text{m}^2$, the maximum coverage area is $4^{21}\,\text{m}^2$.

According to the addressing encoding method, the storage structure of the octree can be encoded. In order to easily record, the code can be converted to decimal representation as shown in Figure 3, which is a twice-segmented 3D scene. The specific steps of segmentation and encoding are as follows:

(1) The 3D map scene is represented in the form of cube approximately.

(2) By treating this cube as the root node of the tree, according to its deviation from the three axes, it is marked with three binary numbers, and then the binary numbers are converted into corresponding decimal numbers. The first segmentation and coding are completed.

(3) After the first segmentation, the regions numbered 0, 2, 3, 5, and 7 do not contain data, so they are white nodes and can be used as leaf nodes without further segmentation. The regions numbered 1 and 6 contain one data, respectively, so they are black nodes and can be regarded as leaf nodes without further segmentation. The region numbered 4 contains 4 data, so it is a gray node. It cannot be regarded as a leaf node, and the segmentation should continue; repeat step (2) to divide and encode.

(4) After the second segmentation, the region originally numbered 4 is divided into 8 subregions, among which the nodes numbered 40, 41, 42, 43, 44, and 47 do not contain data, so they are white. The nodes numbered 45 contain two data, and the node numbered 46 contains two data, so the two nodes are black. At this time, all nodes can be regarded as leaf nodes, so there is no need to segment, and the recursive operation of scene segmentation is finished.

3.1.3. Data Expression. In three-dimensional indoor maps, most of the shapes of objects are very irregular, and the objects with cube shape occupy a much larger proportion than that with spherical shape. Therefore, after statistical

analysis, Axially Aligned Bounding Box (AABB) method was considered to be more appropriate. The enclosing shape is a cuboid, and each side of the cuboid needs to be consistent with the corresponding coordinate axis direction. The benefit of normalizing the cube is that the AABB can be easily calculated based on the maximum and minimum coordinates of each vertex. The size and coordinates are as follows:

$$pos = \{(x, y, z) | x_{\min} \le x \le x_{\max}, y_{\min} \le y \le y_{\max}, z_{\min} \le z \le z_{\max}. \tag{4}$$

The minimum coordinate of point A in the lower-left corner and the maximum coordinate of point B in the upper-right corner can be selected and wrapped into a range. Two points A and B can be used to represent the bounding box as shown in Figure 4.

By combining the above octree scene segmentation, node addressing encoding, and data approximate expression AABB, a storage model based on traditional octree can be formed. However, due to its approximate expression, some error rates still exist and are acceptable.

3.2. Design of Map Data Storage Method. In the process of saving the approximate representation data into the octree, few practical problems appeared such as map data conflict, partition line and object conflict, etc. Therefore, an improved octree structure with backtracking characteristics is proposed in this paper.

3.2.1. Resolution to Map Data Conflict. In the three-dimensional indoor map scene, the distribution of objects is irregular. Many objects are too close to each other and even overlap. Therefore, when using bounding boxes to represent the objects approximately, the bounding boxes may intersect or overlap. As AABB is cube and cube box sides, respectively, on the axis, when a minimum coordinate and a maximum coordinate are used to represent the cube, we should check whether the coordinates of the two objects overlap or not. Suppose the coordinates of A are $\{(x_{A_{\min}}, y_{A_{\min}}, z_{A_{\min}})\}$, $\{(x_{A_{\max}}, y_{A_{\max}}, z_{A_{\max}})\}$, and the coordinates of B are $\{(x_{B_{\min}}, y_{B_{\min}}, z_{B_{\min}})\}$, $\{(x_{B_{\max}}, y_{B_{\max}}, z_{B_{\max}})\}$; then, the judgment conditions are as follows:

(1) If $x_{A_{\max}} < x_{B_{\min}}$ or $x_{B_{\max}} < x_{A_{\min}}$, the bounding box cubes of two objects do not conflict

(2) If $y_{A_{\max}} < y_{B_{\min}}$ or $y_{B_{\max}} < y_{A_{\min}}$, the bounding box cubes of two objects do not conflict

(3) If $z_{A_{\max}} < z_{B_{\min}}$ or $z_{B_{\max}} < z_{A_{\min}}$, the bounding box cubes of two objects do not conflict

If none of the above three conditions is satisfied, the AABB surrounded cubes of two objects are to be in conflict. In the three-dimensional map of indoor scenarios, due to the relatively small volume of a single object, for the conflicted data, we should use a bigger bounding box for approximate expression, that is, obtaining the maximum and minimum coordinates of the four coordinates used by two objects to represent the coordinates of the new bounding box. Because this kind of conflict is not a lot in the actual indoor environment and the new bounding box volume is not large, the cost is acceptable. Another advantage of taking the method is to help solve the difficulty caused by the unexpected situation of the dividing line at the intersection of two objects of the AABB bounding boxes.

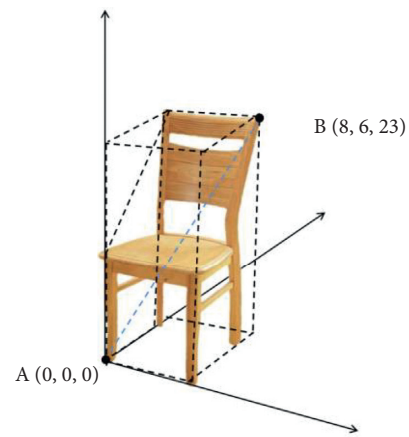


Figure 4: The sample of AABB object.

3.2.2. Resolution to Conflict between Dividing Line and Object. The linear way can replace the storage of pointers by addressing codes, which only need to record the addressing codes of leaf nodes, which can greatly reduce the storage pressure. In addition, the linear octree can be transformed into the corresponding linear list for representation. The elements in the linear list correspond to the nodes, which can increase the convenience of calculation.

Linear mode is one of the most widely used storage methods. All data of this method are only stored in the leaf nodes, and the middle nodes do not store any data. However, in the three-dimensional indoor map scene, the location of indoor data is very close. It often exists in the situation that the dividing line is on the indoor object. If the object data is segmented at this time, it will terminate the object data and need to be reconstructed again. This is a complex process and requires a large amount of calculation. Therefore, this method is not applicable to the data storage of 3D indoor map.

According to the characteristics of the close distribution of indoor map data, a new data storage model is proposed in this paper. It improves the linear octree and makes it to have backtracking ability. Originally, the map data is still stored at the leaf nodes, when the dividing line occurs in the object, the object data is not divided and instead it gets stored in the nearest parent node that can fully contain the object data. Since the object data storage has been completed at this time, the scene segmentation can be continued.

3.2.3. Algorithm Implementation

(1) Definition of relevant data structure

(I) Leaf Tree Node structure

```
Struct LeafTreeNode
{
Coordinate [4]//Coordinates of the cube
NodeCoding//Location code
NodeDepth//The level of node
ObjectNum//The number of objects contained in nodes
ObjectID[objectNum]//Object code
}
(II) Non-Leaf Tree Node structure
Struct Non-LeafTreeNode
{
Coordinate [4]//Coordinates of the cube
NodeCoding//Location code
NodeDepth//The level of node
Children [8]//the pointers point to the node's children
ObjectNum//The number of objects contained in nodes
ObjectID[objectNum]//Object code
}
(III) Object structure
Struct Object
{
ObjectID//Object code
ObjectSize//Object size (AABB)
RoomID//Room code that the object is in
}
(IV) Room structure
Struct Room
{
ObjectID//Object code
ObjectSize//Object size (AABB)
RoomID//Room code the object in
}
```

The three structures defined above have different functions, but they are also related to each other. The algorithm of creating an octree is as follows Algorithm 1.

3.3. Closed-Orientation 3D Indoor Map Data Retrieval Method. Based on the data storage model which is proposed in this paper, a data retrieval method with closed constraints was proposed according to the characteristics of more indoor partition to improve the efficiency of data retrieval.

3.3.1. Analysis of Neighbor Retrieval Algorithm Based on the Octree. In octree structure, there are at most 26 neighbor nodes of a node. According to the type of critical location between the target node and neighbor node, three cases exist: surface-neighbor, edge-neighbor, and point-neighbor, as shown in Figure 5. There are 6 surface-neighbors, 12 edge-neighbors, and 8 point-neighbors for one node.

In the octree, the calculation of neighbor nodes can be realized by finding the nearest common ancestor of target node and neighbor nodes. According to the direction and area information of the nodes, this method completes the calculation of address coding of neighbor nodes. This calculation is based on the modifying of the target node addressing code. The process is from the rightmost coding in turn to the left, updating the value of the target node at that position with the numerical results obtained by each bit of computation.

In the octree environment, if we want to find the 26 direction's information, we can only record the surface-neighbor data in orthogonal direction {D, U, R, L, B, F}. The 12 edge-neighbors and 8 point-neighbors' data can achieve through orthogonal direction data twice or thrice operation. This can reduce the amount of data table.

When the target node and the neighbor node in the given direction are brothers, the calculation method executes the "stop" command and directly modifies the rightmost digit value. Otherwise, the nearest common ancestor of the two nodes should be located, and then relevant calculation can be performed. When the layer of neighbor nodes is deeper than the target nodes, which is due to the neighbor node calculation method can only calculate the same size nodes with the target nodes, and the nodes calculated are not leaf nodes. That is, the nodes adjacent to the target nodes can also continue to refine. Although it is difficult to solve this problem, the research background of this paper is 3D indoor maps. Most of the map data volume is small, and therefore we can regard the calculated neighbor nodes as the target node's neighbor nodes, and it does not need any further detailed analysis. The effects of these errors are negligible.

3.3.2. Neighborhood Retrieval Optimization Algorithm in Indoor Map Scene. In the traditional octree neighbor node retrieval methods, the results of each retrieval are the target node's 26 neighbor nodes in all directions. As we knew from the analysis mentioned earlier, there is no need of all 26 neighbor nodes in the 3D indoor scenarios. We can take the indoor room partition into consideration. The neighbor nodes in the same room with the target nodes are needed with the data. The optimization of the traditional octree neighbor retrieval method can realize the filtrating of neighbor nodes, reduce the number of neighbor nodes, reduce the workload for the later neighbor data analysis, thus reduce the cost of retrieval, and improve the stability of the algorithm.

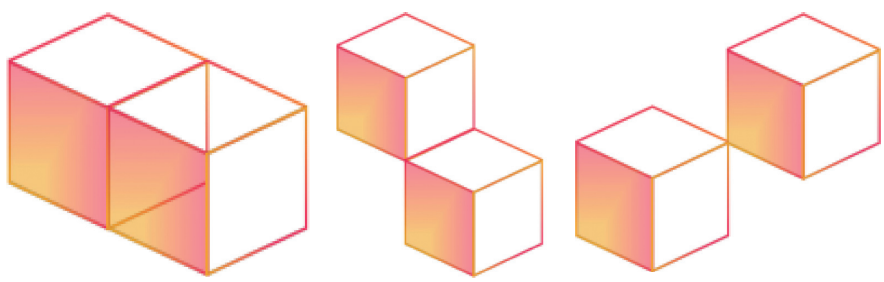
In the actual retrieval process of indoor map, according to the area where the target data is located, it can be divided into the following situations:

(1) The retrieval of data on the room ground: the retrieval in this environment can improve the efficiency by using the closure neighbor retrieval method.

(2) Corridor data retrieval: unlike the data on the room ground, the objects in corridor are limited, the environment is relatively open, and the enclosed space

```
Input: TreeNode//Map scene is as root node
Output: TreeNode//The root node of the segmented data storage model
Begin:
  For i = 0: MAX//Scene segmentation
    newNodeDepth = nodeDepth + 1
    newNodeParents = node
    newNodeAABB = nodeAABB/8
    If nodeAABB.value > 2 and node.AABB > nodeMin then
      If Line on nodeAABB then
        Search for the smallest ancestor
        Data is stored in the ancestor node
        Delete original data
      End If
      Divide node
      Node coding
    End If
  End For
  Store leaf node data
  Return TreeNode
End
```

Algorithm 1: Octree Map Data Storage.



(a) (b) (c)

Figure 5: The neighbor nodes of a node. (a) The surface-neighbor. (b) The edge-neighbor. (c) The point-neighbor.

is few. Therefore, using neighbor search algorithm does not improve the efficiency, and it will lead to unnecessary retrieval costs. So, in this case, we can retrieve directly.

(3) The retrieval of data inside the room of nonground: the data in the area mainly refers to the space among walls and ceiling; it is similar to the data in the corridor. The purpose of retrieving it is not to move it, such as lamps on the ceiling. The purpose of retrieval is to repair, not to change the position. So even though it is closure, it does not meet the usage scenarios of closure neighbor search algorithm; this case also can retrieve directly.

These three cases can contain most of the data, and direct retrieval is sufficient for extreme cases.

Because different retrieval algorithms are used in different scenarios, the logical relationship of the object position parameters is as follows: x represents object data, $A(x)$ represents data on the floor of the room, $B(x)$ represents data on the corridor, and $C(x)$ represents data in the room that is not on the floor:

$$A(x) \longrightarrow \exists x(B(x) \wedge C(x)). \tag{5}$$

N represents neighbor search:

$$\exists x(B(x) \wedge C(x)) \wedge A(x) \longrightarrow N. \tag{6}$$

Object data on the floor of the room requires neighbor retrieval. $N(x, y)$ is the data that x and y are neighbor. $R(x, y)$ is the data that x and y are in the same room:

$$\exists y(N(x, y) \wedge R(x, y)) \wedge \forall y(N(x, y) \longrightarrow R(x, y)). \tag{7}$$

Although there are cases in which neighbor data is in the same room, not all the neighbor nodes are in the same room.

3.3.3. Algorithm Design. Considering the closure of data, the data retrieval method is the combination of the traditional octree neighbor nodes retrieval method and the closure of indoor rooms. The calculated neighbor data is filtered again by adding the constraint condition of room closure:

(1) Compare the ID of the target data with the node data to find the node where the target data is located and

record the addressing code of the node, which is called the target node.

(2) According to the addressing code of the target node, retrieve the location of the target node in the octree and compare the data in the node with the target data ID to obtain the target data.

(3) If the target data do not conform to the application scope of neighbor retrieval, the retrieval shall be stopped; otherwise, the next step shall be carried out.

(4) Calculate the neighbor nodes of the target node and consider the room constraints, no more than 26. Parse the data in the qualified neighbor node. The two algorithms of octree retrieval and octree neighbor retrieval are shown in Algorithms 2 and 3, respectively.

4. Results and Discussion

4.1. Creation of Experimental Scenarios. In order to ensure the standardization and convenience of 3D indoor map scene creation, SketchUp and ArcGIS were used to create the map scene. ArcMap was used to create the two-dimensional scene base map, in which the surface data structure was used as the wall, and the point data structure was used as the marker point of each specific object in the scene. Finally, in the ArcScene platform, we added a height value for each surface type data structure and made it with a 3D data including height value. We changed the symbol expression mode of each point data to the specific object form which it represented, such as computer table, sofa, cabinet, etc. These specific objects needed to be done by SketchUp specific components. For the purpose of beauty, wall, ground, windows, and doors can be decorated with concrete material. The final scene is shown in Figure 6.

Map scenario consists of two floors; there is a corridor in the middle section of each floor. On both sides, there are some rooms of different sizes. The basic object types include computer chair, table, and bookcase inside the room, a total of 87 indoor objects, and the specific parameters of floor 1 and floor 2 as shown in Tables 1 and 2, not considering the dining regions and stairwells.

4.2. Experiments and Analysis of Neighbor Retrieval Algorithm. For the map data retrieval, the time required is an important index to evaluate the efficiency of a retrieval method. Therefore, in the experiment, based on the proposed storage model, the octree data retrieval method, R-tree data retrieval method, and octree neighbor data retrieval method are tested to analyze the time cost by these three methods in retrieving 3D indoor map data. In order to avoid the impact on the experimental results due to the lack of experimental data, the experiment is conducted under two simulation scenarios. The first experiment scene A is the map scene created in Section 4.1, and the second experiment scene B is shown in Figure 7. The modeling method of this scenario is adopted in Section 4.1.

Both experimental scene A and experimental scene B are three-dimensional indoor maps, and the details of them are shown in Table 3. Now in the second experimental scenarios, three retrieval methods are completed. It is now the analysis of the time used complexity of data retrieval under different conditions.

When 3D indoor map data is directly retrieved without any data structure and retrieval method, the time complexity of retrieval is (n^p), where n is the number of data and p is the number of data to be searched. When only using the proposed data storage model for the experimental data storage, without using any search method, as the octree, it cannot directly link to the other required node address based on a node address. Therefore, it needs to do tree traversal for each retrieval; time complexity of the search is $O(2^{3(L-1)})$, where L is the depth of the tree. R-tree data retrieval method is like octree data retrieval method, but due to the overlapping area occurring in the R-tree, it takes extra time to detect if there is a collision of object data in the overlapping area. In an indoor environment, objects are packed more frequently, so retrieval takes more time. If the concept of "searching for a point, points around the searched point are within the searching range" is used at this time, the neighbor data information is retrieved by the traditional neighbor retrieval method of octree. Because of the regularity of the octree partitioning, it can calculate up to 26 neighbors map data around the target data through the formula. Therefore, the time complexity of neighbor retrieval method is $O(1)$.

This experiment simulated five data retrievals and took scene A as an example for analysis. The data retrieved each time were distributed in multiple rooms. The specific retrieval parameters are shown in Table 4.

Figures 8 and 9 show the comparison of retrieval time among R-tree retrieval, the octree retrieval, and the octree neighbor retrieval under two scenarios, respectively. Figures 10 and 11 show the comparison of the octree retrieval and neighbor retrieval methods in terms of frequency times. From the experimental results, it can be concluded that the octree requires less time consumption in data retrieval operation than R-tree, and its stability is better. It indicates that the octree structure is not only better in construction time than R-tree structure, but also has advantages in data retrieval. For the octree structure, the use of neighbor retrieval method is better than not using it completely, which proves that the idea of "searching for a point, points around the searched point are within the searching range" is feasible. This suggests that neighbor nodes data of the target node is significantly useful. It is verified that the octree neighborhood retrieval can improve the overall efficiency without increasing the time efficiency of single retrieval.

4.3. Experiments and Analysis of Data Retrieval Methods Oriented to Closure Constraints. The experiments in this section will analyze from the aspect of retrieval cost, which refers to the number of neighbor nodes other than the target node each time the neighbor retrieval method is adopted. As the retrieved data needs to be parsed and represented each time, there will be more neighbors and the cost will be higher. The experiments still adopt the simulated retrieval data in the previous section to conduct in the experimental scene A and the experimental scene B.

```
Input: objectID, nodeCoding
Output: target data
Begin: target = root
   While (objectID in CurNode and nodeCoding<>targetNode. nodeCoding) do
      targetNode = CurNode.children[the i-th bit of the nodeCoding]
   While length of node objectNum > 0 do
       /* objectNum is array, does not reach the segmentation threshold */
      collect objectData from targetNode
      if collect objectData = targetObjectData then
         call function OctreeMapNeighborDataRetrieval method
      end while
End
```

Algorithm 2: Octree Map Target Data Retrieval.

```
Input: objectID, nodeCoding, objectPosition
Output: Neighbor data
Begin:
   If objectPosition ≠ roomNoGround and objectPosition ≠ corrido then
   /* Retrieval scenario judgment */
      call NeighborRetrieval method
   End If
End
Function NeighborRetrieval
   For i = 1 : 26
      If targetObject.roomID = neighborObject.roomID then/* Room ID judgment */
         Return True
      End If
   End For
End
```

Algorithm 3: Octree Map Neighbor Data Retrieval.

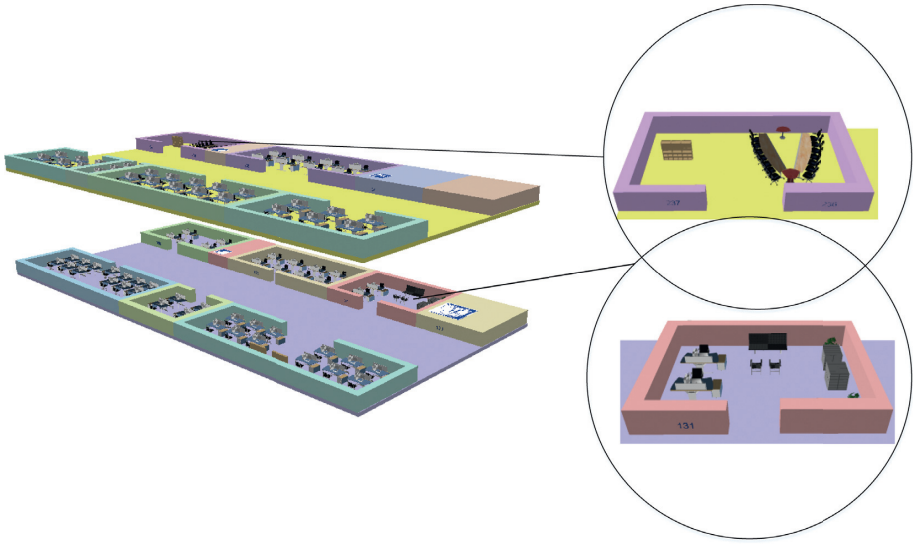


Figure 6: The experimental scenario A.

Table 1: Floor 1 parameters.

Room ID	134	132	131	122	123	124
Object number	5	8	6	13	4	15
Object type	1	1	5	1	1	2
Room length (m)	23	30	28	31	4	55

Table 2: Floor 2 parameters.

Room ID	237	235	225	226	227	228
Object number	2	8	6	3	10	6
Object type	2	1	1	1	1	2
Room length (m)	31	33	30	5	29	26

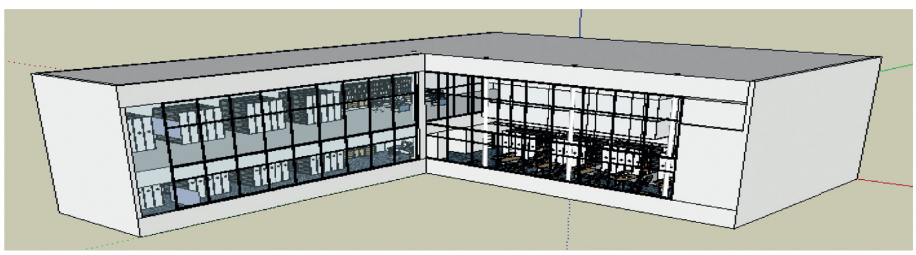


Figure 7: The experimental scene B.

Table 3: Experimental scene parameters.

	Name	Size (kB)	Data format	Object number	Scene length (m)
Scene A	Lab building	16691	sxd	87	90
Scene B	Library	17347	skp	103	60, 46, 90

Table 4: Simulated retrieval operation parameters.

Times	1st	2nd	3rd	4th	5th
Data number	10	15	8	18	20
Room 1 (ID, objects number)	132, 4	122, 8	225, 3	124, 10	227, 5
Room2	131, 4	225, 4	228, 5	123, 4	124, 9
Room3	237, 2	132, 3	—	227, 4	228, 6

From Figures 12 and 13, we can observe that the calculation method of neighbor nodes after adding the closure constraint condition has obviously reduced most cases in the number of neighbor nodes. Compared with the search method without constraint conditions, the search cost is reduced by 25 percent on average, which proves that the room closure condition can reduce the cost of data retrieval and then improve the efficiency of data retrieval on the premise of guaranteeing the efficiency of neighbor retrieval.

5. Conclusions

In this study, we evaluated the problems related to the storage consumption of 3D indoor maps. In order to reduce storage consumption and improve storage efficiency, a 3D data storage model based on the octree structure was proposed. The octree structure is used to segment the indoor map data. The addressing code was used to uniquely mark the generated nodes without retaining the pointers to the parent and child nodes. Compared with pointer storage, this storage method saved 60% of storage space. Considering the complexity caused by irregular shape of indoor objects, AABB method was used to approximately express the object data in the map scene. The advantage of this storage model based on scene segmentation lied in the fact that it could reduce the amount of storage units and improve the storage efficiency. Moreover, the connection between each storage unit was convenient for future management and use. Then, according to the characteristics of close distribution of indoor map data, the situation of octree scene segmentation

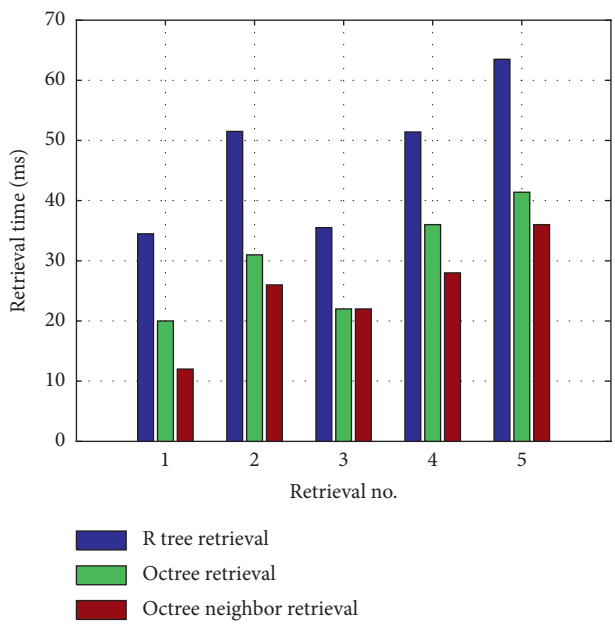


Figure 8: The simulated data retrieval in scenario A.

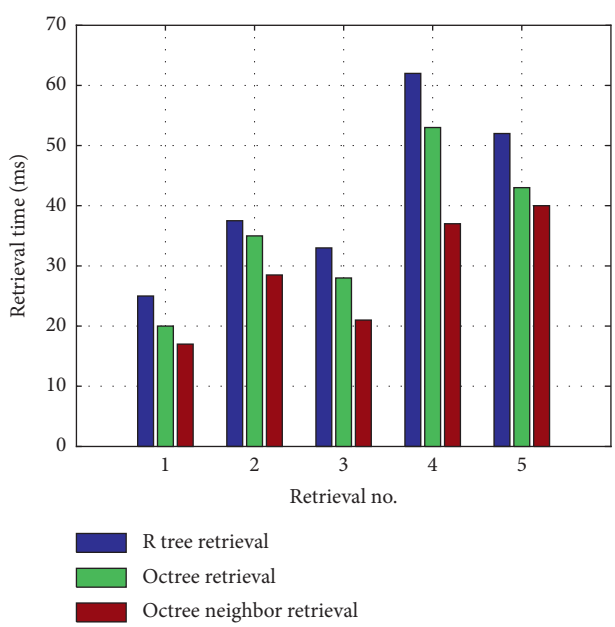


Figure 9: The simulated data retrieval in scenario B.

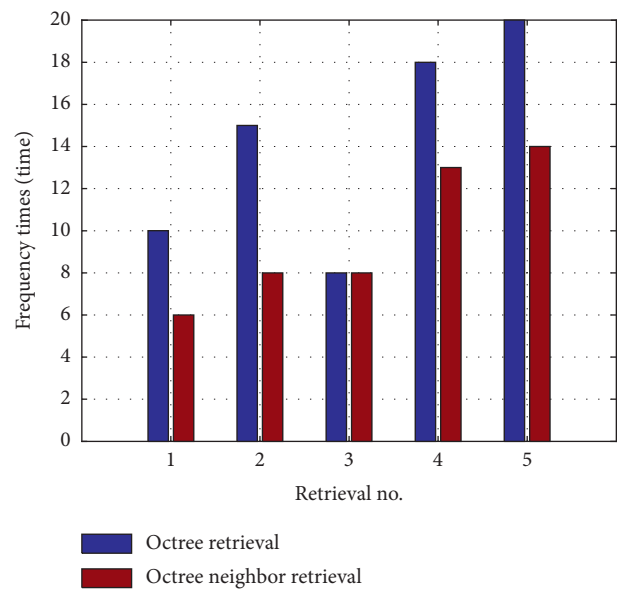


Figure 10: The number of data retrievals in scene A.

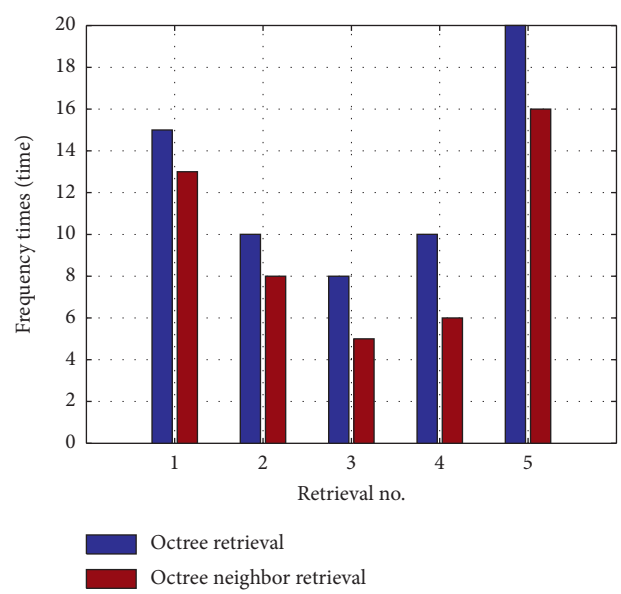


Figure 11: The number of data retrievals in scene B.

line within the object data range was analyzed. The linear octree was improved to have a backtracking ability. When the dividing line occurred in the object, the object data was not divided and instead it got stored in the nearest parent node that could fully contain the object data, not just in leaf nodes. Ultimately, it led to strengthening the stability of data storage and improving the efficiency of storage. After experimental verification, the data storage model of the octree map did not consume much time compared with the traditional octree, which is 1/8 of the time consumed by R-tree structure. Moreover, considering the problem of high cost and low efficiency of data retrieval, a data retrieval method based on the octree storage model was proposed in this paper. The method mainly analyzed from two aspects. Initially, according to the feature of "searching for a point, points around the searched point are within the searching range" of map data retrieval, an octree neighbor nodes retrieval method was proposed to reduce the number of retrieval operations. Then, by adding closure constraint conditions, the neighbor nodes retrieval method was improved to decrease the cost of data retrieval. Then, we could improve the efficiency of data retrieval. Experimental results showed that the cost of retrieval was reduced by about 25%.

Due to the huge amount of indoor map data, the consumption of storage is crucial to the promotion and application of the system. The index linear storage based on octree

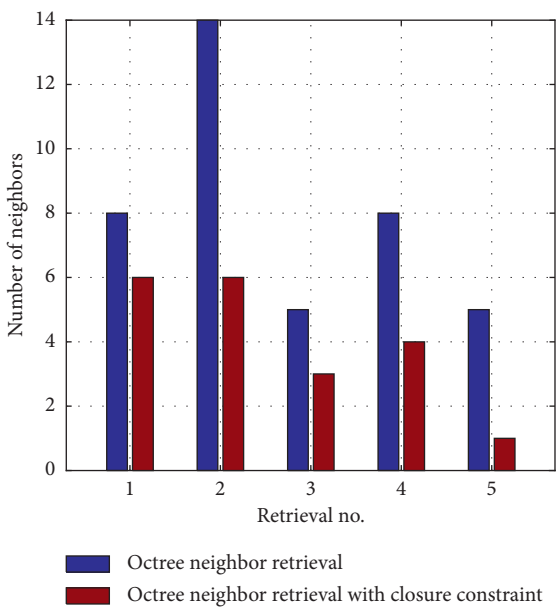


Figure 12: The number of neighbors in scene A.

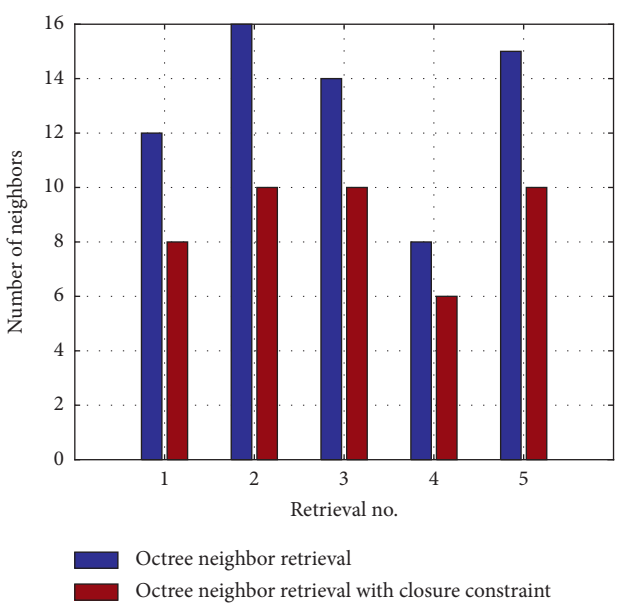


Figure 13: The number of neighbors in scene B.

can save a lot of storage space, reduce storage costs by 60%, and save system development costs. Meanwhile, the neighbor retrieval method proposed by adding closed constraints improved the retrieval efficiency by about 25%. The method of map data storage and retrieval proposed in this paper saved storage space and improved retrieval speed in practical applications. The indoor map was used in indoor localization to display the positioning results in indoor scenes, so that users would have a good positioning experience. With the advent of the 5G era, indoor localization has developed rapidly, and the search efficiency of indoor maps will be increasingly demanded. This research can achieve the purpose of real-time display of positioning results in map scene.

There are many potential future directions existing for this type of work. First, in terms of map data expression, AABB packaging box approximate expression method was adopted in this paper, which can meet current needs. However, for objects with special shapes, some segmentation methods based on point-cloud data can be adopted to improve the accuracy of data expression. Second, there are vast amounts of data and many concurrent operations such as data retrieval in the indoor map. Therefore, cloud storage method and distributed system architecture can be combined to improve the robustness of data storage and retrieval, which could enhance the application experience of users.

Data Availability

The datasets used and analyzed during the current study are available from the first author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (no. 61872104), the National Science and Technology Major Project of China (no. 2016ZX03001023-005), the China Postdoctoral Science Foundation (no. 2019M651264), the Natural Science Foundation of Heilongjiang Province of China (no. 2015023), and the Basic Business Project in Education Department of Heilongjiang Province of China (nos. 135109243 and 135209237).

References

[1] L. Jiang, L. D. Xu, H. Cai et al., "An IoT-oriented data storage framework in cloud computing platform," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1443–1451, 2014.

[2] S. Bemel-Benrud, T. C. MacWright, E. Halperin et al., "Providing visual selection of map data for a digital map," U. S. Patent Application: US20180052593A1, 2018.

[3] S. Har-Noy, L. Barrington, and N. Ricklin, "System and method for large scale crowdsourcing of map data cleanup and correction," U. S. Patent Application: US20160004724A1, 2016.

[4] H. Lu and M. A. Cheema, "Indoor data management," in *Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering*, IEEE, Helsinki, Finland, pp. 1414–1417, May 2016.

[5] S. Malhotra, M. N. Doja, B. Alam, and M. Alam, "Skipnet-octree based indexing technique for cloud database management system," *International Journal of Information Technology and Web Engineering*, vol. 13, no. 3, pp. 1–13, 2018.

[6] Y. Guo, "Method and device for constructing spatial index of massive point cloud data," U.S. Patent Application: US20180081034A1, 2018.

[7] Y. Wanqiang, Z. Junliang, C. Peng et al., "An octree-based mesh simplification algorithms for 3-dimension cloud data," *Science of Surveying and Mapping*, vol. 41, pp. 18–22, 2016.

[8] R. Kothuri and S. Ravada, "Oracle spatial, geometries," *Encyclopedia of GIS*, pp. 1–9, Springer, Berlin, Germany, 2015.

[9] L. I. Zhen-Ju, L. I. Xue-Jun, Y. Sheng et al., "Spatial index built in cloud computing environment," *Geomatics & Spatial Information Technology*, vol. 38, no. 10, pp. 13–17, 2015.

[10] Z. X. Qiao, "Application and research on spatial index technology," *China Place Name*, vol. 6, pp. 72-73, 2014.

[11] C. Yang, "Research on fast index technology of spatial data," *Electronic Technology & Software Engineering*, vol. 19, pp. 207-208, 2018.

[12] Y. Xiao, J. T. Li, W. J. Zhang et al., "A spatial index structure of natural neighbor relationship query," *Geomatics World*, vol. 25, no. 1, pp. 32–38, 2018.

[13] Y. Yang, S. Papadopoulos, D. Papadias, and G. Kollios, "Authenticated indexing for outsourced spatial databases," *The VLDB Journal*, vol. 18, no. 3, pp. 631–648, 2009.

[14] W. Wang, Z. Xuan, L. Sun, Z. Jiang, and J. Shang, "BRLO-tree: a data structure used for 3D gis dynamic scene rendering," *Cybernetics and Information Technologies*, vol. 15, no. 4, pp. 124–137, 2015.

[15] M. H. Namdari, S. R. Hejazi, and M. Palhang, "MCPN, octree neighbor finding during tree model construction using parental neighboring rule," *3D Research*, vol. 6, no. 3, p. 29, 2015.

[16] W. Solihin, C. Eastman, and Y. C. Lee, "Multiple representation approach to achieve high-performance spatial queries of 3D BIM data using a relational database," *Automation in Construction*, vol. 81, pp. 369–388, Article ID S0926580517302388, 2017.

[17] P. F. Wang, Z. Y. Ding, A. G. Teng et al., "Research on the application of quadtree index in Jiangsu power grid GIS platform," *Power Information and Communication Technology*, vol. 15, no. 7, pp. 87–91, 2017.

[18] D. J. R. Meagher, "The octree encoding method for efficient solid modeling," Technical Report IPL-TR-032, Image Processing Lab. Rensselaer Polytechnic Institute, New York, NY, USA, 1982.

[19] D. Meagher, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing*, vol. 19, no. 2, pp. 129–147, 1982.

[20] S. Sengupta and P. Sturgess, "Semantic octree: unifying recognition, reconstruction and representation via an octree constrained higher order MRF," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Seattle, WA, USA, pp. 1874–1879, May 2015.

[21] Y.-T. Su, J. Bethel, and S. Hu, "Octree-based segmentation for terrestrial LiDAR point cloud data in industrial applications," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 113, pp. 59–74, 2016.

[22] P. Beno, V. Pavelka, F. Duchon et al., "Using octree maps and RGBD cameras to perform mapping and A*navigation," in *Proceedings of the 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, pp. 66–72, IEEE, Ostrava, Czech Republic, September 2016.

[23] J. Yun, Y. S. Lee, and Y. Jung, "Development of a gap searching program for automotive body assemblies based on a decomposition model representation," *Advances in Engineering Software*, vol. 81, pp. 7–16, 2015.

[24] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: learning deep 3D representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 3, Honolulu, HI, USA, July 2017.

[25] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: efficient convolutional architectures for high-resolution 3d outputs," 2017, https://arxiv.org/abs/1703.09438.

[26] Z. Deng, L. Wang, W. Han et al., "G-ML-octree: an update-efficient index structure for simulating 3D moving objects across GPUs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 5, 2017.

[27] D. Taniar and W. Rahayu, "A taxonomy for nearest neighbour queries in spatial databases," *Journal of Computer and System Sciences*, vol. 79, no. 7, pp. 1017–1039, 2013.

[28] C. H. Chen, "A cell probe-based method for vehicle speed estimation," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. 1, no. E130A, pp. 265–267, 2020.

[29] X. Chen, N. Liu, Y. Su, and G. Chen, "A survey of swarm intelligence techniques in VLSI routing problems," *IEEE Access*, vol. 8, no. 8, pp. 26266–26292, 2020.

[30] G. G. Liu, Z. S. Chen, Z. Zhuang et al., "A unified algorithm based on HTS and self-adapting PSO for the construction of octagonal and rectilinear SMT," *Soft Computing*, vol. 24, pp. 3943–3961, 2020.

[31] M. Pan, Y. Liu, J. Cao et al., "Visual recognition based on deep learning for navigation mark classification," *IEEE Access*, vol. 99, p. 1, 2020.

[32] M. Pan, J. H. Zhou, H. Cao et al., "Water level prediction model based on GRU and CNN," *IEEE Access*, vol. 8, no. 8, pp. 60090–60100, 2020.

[33] W. X. Li, C. Y. Zhang, G. G. Liu et al., "Extraversion measure for crowd trajectories," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 15, pp. 6334–6343, 2019.

[34] C. H. Chen, F. J. Hwang, H. Y. Kung et al., "Travel time prediction system based on data clustering for waste collection vehicles," *IEICE Transactions on Information and Systems*, vol. 7, no. E102D, pp. 1374–1383, 2019.

[35] C.-H. Chen, "An arrival time prediction method for bus system," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4231-4232, 2018.

[36] S. Cho, S. Park, G. Cha, and T. Oh, "Development of image processing for crack detection on concrete structures through terrestrial laser scanning associated with the octree structure," *Applied Sciences*, vol. 8, no. 12, p. 2373, 2018.

[37] B. Ummenhofer and T. Brox, "Global, dense multiscale reconstruction for a billion points," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1341–1349, Santiago, Chile, December 2015.

[38] F. Steinbrücker, J. Sturm, and D. Cremers, "Volumetric 3D mapping in real-time on a CPU," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Hong Kong, China, pp. 2021–2028, May 2014.

[39] W. Jinqiu, Q. Gang, and K. Pengbin, "Emerging 5G multi-carrier chaotic sequence spread spectrum technology for underwater acoustic communication," *Complexity*, vol. 2018, Article ID 3790529, 7 pages, 2018.

[40] C. H. Chen, F. Y. Song, F. J. Hwang et al., "A probability density function generator based on neural networks," *Physical A: Statistical Mechanics and Its Applications*, vol. 3, no. 541, Article ID 123344, 2020.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6591873, 10 pages
https://doi.org/10.1155/2020/6591873

Research Article

Vehicle Logo Recognition with Small Sample Problem in Complex Scene Based on Data Augmentation

Xiao Ke [1,2,3] **and Pengqiang Du** [1,2,3]

[1] *College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China*
[2] *Fujian Provincial Key Laboratory of Networking Computing and Intelligent Information Processing (Fuzhou University), Fuzhou 350116, China*
[3] *Key Laboratory of Spatial Data Mining & Information Sharing, Ministry of Education, Fuzhou 350003, China*

Correspondence should be addressed to Pengqiang Du; fzu_pengqiangdu@outlook.com

Received 1 April 2020; Revised 22 May 2020; Accepted 9 June 2020; Published 9 July 2020

Guest Editor: Feng-Jang Hwang

Automatic identification for vehicles is an important topic in the field of Intelligent Transportation Systems (ITS), and the vehicle logo is one of the most important characteristics of a vehicle. Therefore, vehicle logo detection and recognition are important research topics. Because of the problems that the area of a vehicle logo is too small to be detected and the dataset is too small to train for complex scenes, considering the speed of recognition and the robustness to complex scenes, we use deep learning methods which are based on data optimization for vehicle logo in complex scenes. We propose three augmentation strategies for vehicle logo data: cross-sliding segmentation method, small frame method, and Gaussian Distribution Segmentation method. For the problem of small sample size, we use cross-sliding segmentation method, which can effectively increase the amount of data without changing the aspect ratio of the original vehicle logo image. To expand the area of the logos in the images, we develop the small frame method which improves the detection results of the small area vehicle logos. In order to enrich the position diversity of vehicle logo in the image, we propose Gaussian Distribution Segmentation method, and the result shows that this method is very effective. The $F1$ value of our method in the YOLO framework is 0.7765, and the precision is greatly improved to 0.9295. In the Faster R-CNN framework, the $F1$ value of our method is 0.7799, which is also better than before. The results of experiments show that the above optimization methods can better represent the features of the vehicle logos than the traditional method, and the experimental results have been improved.

1. Introduction

The core idea of intelligent traffic [1] is to use sensors, cameras, and other ways to collect traffic data and use computer-aided management to replace the traditional manual monitoring. For example, we can track moving objects in the video [2, 3] and analyze the salient objects [4, 5] to find traffic abnormalities. It can quickly complete traffic data analysis [6], sharing and retrieval, to achieve the integration of traffic management. Computer vision technology plays a vital role in this field. For a vehicle, its license plate and logos are two important pieces of information. There are various researches on license plate recognition [7, 8], and numerous mature and stable systems have been developed and widely used in actual production scenarios. On the contrary, the research of vehicle logo recognition is not enough. And this is an important part of intelligent transportation; therefore, the demand for vehicle recognition will be increasing in the future. There have been some studies on the recognition of the vehicle logo. Yu et al. [9] proposed a Bag-of-Words-based method for vehicle logo classification. Firstly, dense-SIFT (dense-SIFT) feature extraction is performed on the vehicle logo. After feature extraction, these features are divided into k clusters. Then, the histograms of each image corresponding to these clusters are counted and described. The feature of the whole image is finally trained and classified by support vector machine (SVM). Cyganek et al. [10] put forward a vehicle recognition method based on Tensor Representation in literature. This

method establishes a corresponding 4D tensor classifier for each class of vehicle logos and achieves the function of classifying each vehicle logo by combining the established classifiers. However, this method also locates the vehicle logo by detecting the prior position of the license plate. Llorca et al. [11] attempted to use a variety of features for the vehicle logo and finally decided to construct the system in the form of gradient histogram (HOG) features and support vector machine (SVM). Peng et al. [12] proposed a statistical random sparse distribution method to extract the features of the vehicle logo and then classify the extracted features by the classifier. This method is optimized for the vehicle logo, so it is more robust than some general features such as HOG and SIFT. Huang et al. [13] proposed a vehicle recognition system based on pretraining strategy. Detection and recognition are achieved by combining traditional detection methods with the recognition of convolutional neural networks. In order to reduce the false detection error caused by the traditional detection method, the system expands the detection range. Soon et al. [14] proposed a new pretreatment method. Before using the deep neural network method to extract features, they use the white transformation technology to process the image, removing the pixel redundancy of the adjacent image, making the characteristics of the vehicle logos more obvious.

In previous studies, many methods of vehicle logo classification have been proposed [15], but the location of the vehicle logo has always been a difficult problem in Intelligent Transportation Systems. In the conventional method, the location needs some prehypothesis, such as using the prior position of the license plate or using the characteristics of texture features of headlights to locate the vehicle logo [16], which is not robust to complex situations. However, at present, many deep networks have been able to automatically learn the features of the vehicle logo from the image, thereby directly detecting and recognizing the vehicle logos. But there are also some problems in the application of deep neural networks. The parameters of deep neural networks are very large. At the same time, because of the complexity of its construction, there are many hyperparameters, so it is very difficult to optimize them [17]. When the size of the dataset is too small, the performance of the network will be worse. In order to make the deep neural network better learn and extract the feature representation, we adopt a variety of data optimization methods to make the network perform better in the data of vehicle logos.

In our research, we put forward a new research idea. Compared with the traditional methods, we propose a data optimization method based on the combination of multiple strategies, which is based on the location diversity and scale diversity of vehicle logo.

2. Related Work

The method proposed in this paper is based on the existing deep learning method. Therefore, our research needs a network framework with good performance. We choose the two most representative frameworks, YOLO and Faster R-CNN, as our research objects. At the same time, we need Nonmaximum Suppression algorithm to optimize the detection results.

2.1. YOLO. YOLO algorithm is one of the most popular target detection algorithms [18–20]. Its main advantage is that it can achieve real-time detection, but also get better detection results. The main idea of YOLO algorithm is to divide the image and then directly carry out regression detection on the divided area. It realizes end-to-end detection by inputting images directly into the network to predict. Compared with the existing target detection system, YOLO has a general performance in locating the target area, but it has a better inhibition on the false-positive area of the background.

YOLO detects objects by dividing the image into grids. For the object in the image, when its center is in a grid, the grid is responsible for detecting the object. For these grids, each grid predicts several rectangular boxes that may contain objects, each of which corresponds to a confidence score. This confidence reflects the quality of the prediction of objects for each bounding box. The higher the score, the closer the box will be to the ground truth. If the rectangular box does not contain objects, the confidence should be zero. For each rectangular box, there are five parameters after the prediction: the value of the X and Y coordinates of the object center relative to the grid, the ratio of the rectangular length to the width of the image, and the confidence score mentioned above. In addition, each grid also predicts several conditional probabilities. These probabilities are used to determine the categories of objects, and the number is equal to the number of categories in the dataset.

For the experiment in this paper, five boxes are predicted in each grid. Each box contains four coordinate numbers, one confidence and 30 category probabilities, so each grid has $5 * (5 + 20) = 175$ output dimensions.

2.2. Faster R-CNN. Before the advent of Faster R-CNN, the target detection algorithms proposed by researchers need to use time-consuming region recommendation algorithms to infer the target region [21, 22]. Although some researchers have taken various optimizations to the algorithm, the time consuming of the algorithm is still very huge. In this context, researchers put forward the idea of using CNN directly to predict the target area, and RPN algorithm was born [23].

Faster R-CNN consists of four parts: Conv layer, RPN, ROI Pooling, and Classifier. Conv layer is a convolutional neural network, which is mainly used to extract the features of the original image. It can use VGG [24] or ResNet [25]. The extracted feature map will be used for the RPN layer and full connection layer. RPN is the core idea of Faster R-CNN. It is a full convolution network used to generate region proposals. Firstly, the feature image obtained by Conv layer is further convoluted, and the anchors are generated for each pixel on the image. After that, these anchors are classified by softmax function to determine whether they are the target area. At the same time, anchors are modified by bounding box regression to get more accurate proposals. The Roi Pooling layer is relatively simple. It uses the feature map from Conv layer and the proposals from RPN to form a fixed-size proposal feature map which is then sent to the full connection layer for target recognition and location. Finally, the Classifier sends the proposed feature map from the Roi Pooling layer to the fully

connected network and uses the softmax function to classify. At the same time, L1 loss is used to complete the bounding box regression to obtain the exact position of the object.

2.3. Nonmaximum Suppression. Nonmaximum Suppression [26] (NMS) is very important in the field of computer vision. At present, many mature technologies, such as face detection [27] and pedestrian detection [28], are applied. In the process of object detection, multiple rectangular boxes are often generated. How to filter these candidate boxes to best make the final detection effect has become a problem, and the NMS can be used to complete this task.

The core idea is to compare the confidence of the two candidate boxes and discard the smaller part when the overlap of the two candidate boxes (IOU) is greater than a certain threshold. Loop like this until the IOU of any two candidate boxes is less than the threshold value. At this point, the confidence of the remaining candidate box is the highest, and the detection effect is the best.

3. Proposed Method

3.1. Augmentation and Its Significance. In the field of deep learning, big data is the basis to support the learning of the features of objects [29]. Training a network requires a large amount of data as a support to better extract the features of the targets. If the data quality in a dataset is not good enough, data equilibrium [30] and data augmentation are usually used to optimize it. The significance of data augmentation is to transform the training data and generate new data by certain methods. Through data augmentation, the original dataset can be optimized and expanded. It can prevent the overfitting [31] caused by the small amount of data in the training process. It is of great significance to the recognition and detection ability of the model. For the current vehicle logo datasets, most of them have separated the logos from the scenes, used for classification of vehicle logo images. This kind of dataset cannot be used for training. However, the scale of the dataset labelled directly on the original image is too small to meet the amount of data needed to train. Therefore, data augmentation should be performed on the dataset to meet the needs.

3.2. Traditional Data Augmentation Methods

3.2.1. Rotation. The pixels in the image rotate randomly around the center at the same angle.

3.2.2. Flipping. The pixels symmetrical with horizontal or vertical centerlines are exchanged, which causes the entire image to be upside down or left to right.

3.2.3. Brightness. The overall brightness of the image is promoted or reduced.

3.2.4. Contrast. The difference between the brightest pixel and the darkest pixel in the image is enlarged or narrowed, and the pixel values between them are transformed accordingly.

3.2.5. Noise. The RGB channel value of each pixel in the image is randomly transformed in a proportional, and noise is introduced to cause the image to change.

3.2.6. Cropping. The upper, lower, left, and right boundaries of the image plane are cut with a certain proportion or a certain pixel width.

Many augmentation methods have been proposed so far, but not all augmentation methods are applicable to the vehicle logo dataset. The augmentation methods described above can be broadly divided into two categories: pixel-level operations on the original RGB channel such as brightness transformation and noise disturbance, and no pixel-level operations such as cropping transformation. Through the use of these data augmentation methods, the image becomes more diversified, so that the network can learn the image feature representation more fully.

3.3. Cross-Sliding Segmentation Method. For convolutional neural networks, because of the huge parameters, the amount of data needed is also very large, so we need to find a simple and fast way to generate a large number of effective data. Because the logo area is generally small, it is not easy to segment the logo area when the image is segmented. At the same time, because the location of vehicle logo is very variable, cross-sliding segmentation is used to quickly increase the number of effective images. This method is very fast and easy to implement.

We scan and intercept the image using a rectangular box with a length and width of 1/2 of the source image as Figure 1. This method greatly increases the amount of data. Because the area of the logo is small, it is not easy to split the ground truth boxes, which results in invalid data. The specific segmentation steps are as follows.

Firstly, the original image is divided into 9 subimages, and the length and width of each subimage are 1/2 of the original image:

$$\begin{cases} nw = \dfrac{1}{2} * ow, \\ nh = \dfrac{1}{2} * oh. \end{cases} \tag{1}$$

The position of the generated image in the original image is

$$\begin{cases} xx1 = \dfrac{1}{4} * nw * i, \\ yy1 = \dfrac{1}{4} * nh * j, \\ xx2 = xx1 + nw, \\ yy2 = xx2 + nh. \end{cases} \tag{2}$$

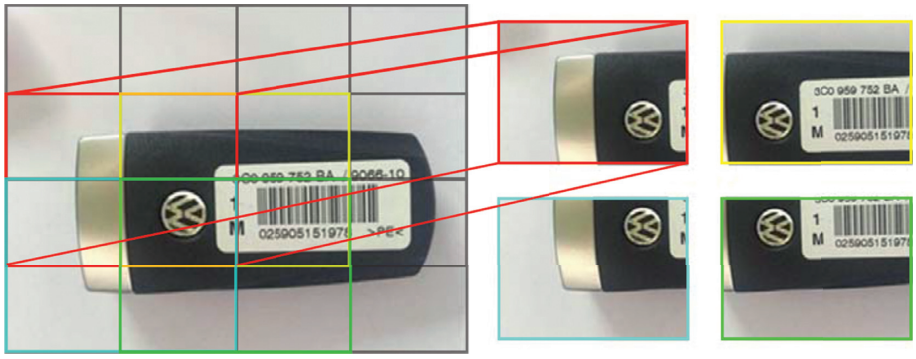


Figure 1: The image to be split. The image above is divided into nine, four of which contain the complete target area.

Among them, $i = 0, 1, 2$; $j = 0, 1, 2$; $xx1$ and $yy1$ are the positions of the upper left corner of the generated image in the original image; $xx2$ and $yy2$ are the positions of the generated image in the lower right corner of the original image; nw and nh are the width and height of the generated image, respectively; ow and oh are the width and height of the original image.

All the generated subimages form set u, while the images that can be used for training form set A. If the target area is r, the subimage that can be used for training should contain r, so

$$A = \{u \mid u \subseteq U, r \subseteq u\}. \tag{3}$$

Cross-sliding segmentation method produced a total of 9 images on Figure 1, 4 of which contain complete vehicle logo information for training, and the remaining should be discarded directly because they do not contain the vehicle logo information or the vehicle logo information is incomplete.

3.4. Small Frame Method. In general images, the proportion of vehicle logos in the whole image is very small, and the background information is too complex, which leads to the lack of main information. At the same time, because the background information is too much, the deep neural network will encounter a lot of noise in the learning process, causing the convergence speed to be too slow.

Therefore, how to suppress the background and highlight the characteristics of the logo has become a difficult problem. In order to make the vehicle logo contribute more information to the whole image, it is necessary to remove the area without vehicle logo. A very easy and effective way is to directly expand the proportion of the logo size in the image.

In this paper, a small rectangular box of random size is used to select the vehicle logo, and other areas without the vehicle logo are discarded to highlight the vehicle logo, so that the network can better extract the features of the vehicle logos.

The position of the generated image in the original image is

$$\begin{cases} xx1 = x1 - \text{random}(1.5,\ 3) * rw, \\ yy1 = y1 - \text{random}(1.5,\ 3) * rh, \\ xx2 = x2 + \text{random}(1.5,\ 3) * rw, \\ yy2 = y2 + \text{random}(1.5,\ 3) * rh. \end{cases} \tag{4}$$

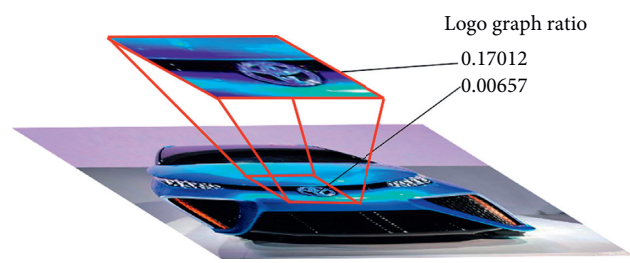


Figure 2: There is an image processed by the small frame method. The vehicle logo accounts for a larger proportion of the graph after processing.

Among them, $xx1$ and $yy1$ are the positions of the upper left corner of the generated image in the original image; $xx2$ and $yy2$ are the positions of the generated image in the lower right corner of the original image; $x1$ and $y1$ are the positions of the upper left corner of the target area in the original image; $x2$ and $y2$ are the positions of the target area in the lower right corner of the original image; the function random$(1.5,\ 3)$ is a function that randomly generates a random number between 1.5 and 3; rw and rh are the width and height of the target area.

Definition 1. The logo graph ratio is the ratio of the area of the logo ground truth to the area of the whole image.

The average graph ratio can measure the situation of the target region in the overall image dataset. The vehicle logo is a small target in the field of object detection, and its graph ratio is generally low:

$$\text{avgRatio} = \frac{1}{N} \sum_{i}^{N} \frac{\text{Area}(ci)}{\text{Area}(oi)}. \tag{5}$$

Among them, avgRatio is the average of all logo graph ratios and Area(ci) is the area size of the logo in image I and Area(oi) is the area size of image I.

Using the method mentioned above, the logo graph ratio of Figure 2 increases from 0.00657 to 0.17012, and the average logo graph ratio of enhanced dataset increases from 0.0144 to 0.0526, about 3.65 times.

3.5. Gaussian Distribution Segmentation Method. Because of the existence of pooling operations in convolutional neural networks, the robustness of the networks to data translation

is very poor. Some studies have shown that, for the same image and the same network, as long as the input image is slightly modified or shifted by one pixel, the output of CNN will change dramatically, and the deeper the network layer is, the more likely this error will occur. The researchers think that there is a certain photographic bias in the general dataset. On the macro level, as long as it is not pixel-level coding, there are no two identical images in the world, so the neural network cannot learn the strict translation invariance and does not need to learn.

The position of the vehicle logo in the image is very variable, so there is a high demand for the diversity of the position of the target in the training set. When the dataset is small, we must optimize the location of the vehicle logo in the dataset. Therefore, we propose the Gaussian Distribution Segmentation method. After the dataset is optimized by this method, the position of the target region will be Gaussian distribution. The size of the generated image is

$$\begin{cases} nw = \max(ox, ow - ox), \\ nh = \max(oy, oh - oy). \end{cases} \tag{6}$$

Among them, nw and nh are the width and height of the generated image; ox and oy are the X coordinate and Y coordinate of the target region center of the original image; ow and oh are the width and height of the original image.

We need to build a function and the function is used to generate a number between 0 and 1, which should meet the standard normal distribution:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{(-x^2/2)}. \tag{7}$$

The coordinates of the target center point of the generated image are

$$\begin{cases} nx = \dfrac{ow}{2} + (nw - rw) * \mathrm{randn}(0, 1), \\ ny = \dfrac{oh}{2} + (nh - rh) * \mathrm{randn}(0, 1). \end{cases} \tag{8}$$

Among them, nx and ny are the X and Y coordinates of the target center point of the generated image; ow and oh are the width and height of the original image; nw and nh are the width and height of the generated image; rw and rh are the width and height of the target rectangle; and $\mathrm{randn}(0, 1)$ is a function that generates a number between 0 and 1 satisfying the Gaussian distribution.

The position of the generated image in the original image is

$$\begin{cases} xx1 = ox - nx, \\ yy1 = oy - ny, \\ xx2 = xx1 + nw, \\ yy2 = xx2 + nh, \end{cases} \tag{9}$$

where $xx1$ and $yy1$ are the positions of the upper left corner of the generated image in the original image; $xx2$ and $yy2$ are the positions of the generated image in the lower right corner of the original image; ox and oy are the X coordinate and Y coordinate of the target center of the original image; nx and ny are the X and Y coordinate of the target center of the generated image; nw and nh are the width and height of the generated image.

4. Results and Discussion

4.1. Experiment Setup. The data in this paper are from the Big Data and Computational Intelligence Competition (BDCI). There are 1131 images of different sizes with annotations totaling 30 categories. And we use 1006 images in its test dataset as the test set of this experiment. The distribution of classes is shown in Figure 3.

Because of the need for data quantity, we augment the original data to expand the dataset. As a contrast, we have experimented with a variety of traditional augmentation methods and the methods proposed in this paper. We try our best to control the amount of data generated by various methods at the same level to ensure the accuracy of the experiment. For these methods, we divide them into two categories: pixel-level operation and non-pixel-level operation. Among them, pixel-level operations include brightness transformation, contrast transformation, and noise addition. Non-pixel operations include cropping, rotation, and flipping. Our own methods based on logo data optimization are all non-pixel-level methods. The amount of data generated by all methods is 2-3 times that of the original dataset. The data distribution generated by these methods is shown in Figure 4.

4.2. Experimental Results. In this paper, three parameters, P, R, and $F1$, are used as the criteria for judging and evaluating the experimental process. Their calculation methods are as follows:

$$\begin{aligned} R &= \frac{1}{N} \sum_{i}^{N} \frac{\mathrm{Correct}(i)}{\mathrm{Ground}(i)}, \\ P &= \frac{1}{N} \sum_{i}^{N} \frac{\mathrm{Correct}(i)}{\mathrm{Propersal}(i)}, \\ F1 &= \frac{2 * P * R}{P + R}. \end{aligned} \tag{10}$$

Among them, $\mathrm{Correct}(i)$ is the number of logos correctly detected, $\mathrm{Propersal}(i)$ is the total number of logos detected, and $\mathrm{Ground}(i)$ is the actual number of logos.

We study the performance of various data optimization methods under the framework of YOLO. By adjusting the parameters and the number of iterations, we get the optimal results under the YOLO framework. First of all, the original data is tested, and the loss changes and data results are as shown in Figure 5.

We trained 50000 iterations under the framework of YOLO and output the results every 1000 iterations. Observing the experimental results, we can find that there is a sudden drop in the loss value between the 10000 and 20000 iterations. The reason is that in this place we use the strategy of learning rate decay, which can make the loss worth further decline. At the same time, it can be seen from (b) that $F1$ value tends to be stable before 50000 iterations, so we can get

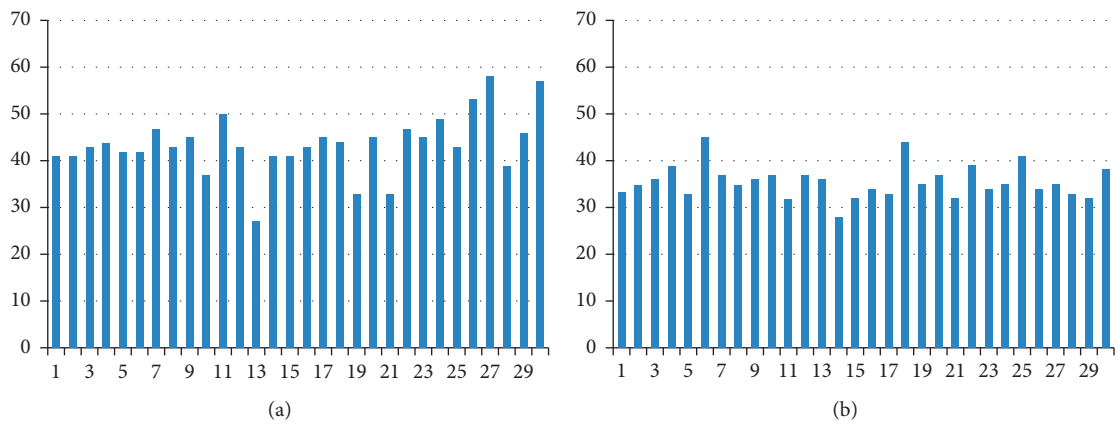


Figure 3: Image distribution of each class. (a) Distribution of training samples. (b) Distribution of test samples.

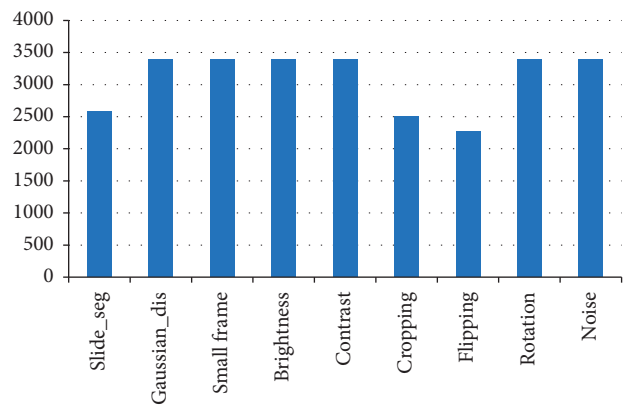


Figure 4: The distribution of new images. By data augmentation, the size of the dataset is expanded to 28859.

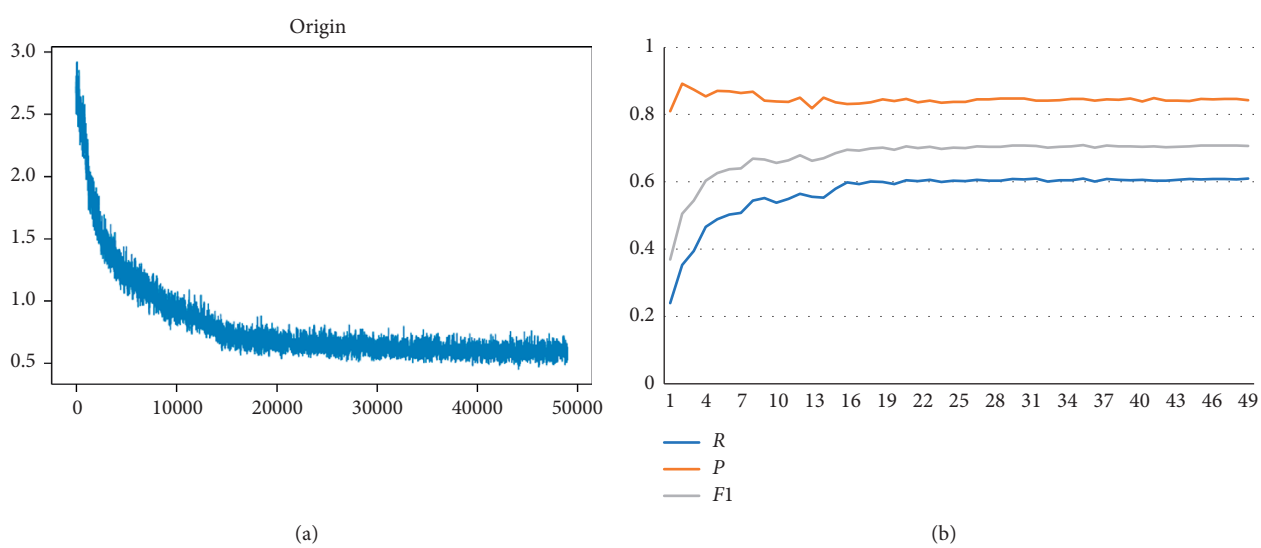


Figure 5: Results of original experiment. We can see that the loss value is stable after 20000 iterations. (a) Loss curve. (b) Curve of R value, P value, and $F1$ value.

an optimal model in 50000 iterations of training. The best results of training on the original dataset are shown in Table 1.

After that, we tried to add the optimized data to the network for training. We experimented with nine enhancement methods, and their loss values changed as shown in Figure 6.

Table 1: The best detection result of the original dataset on YOLO.

P	R	$F1$
0.8466	0.6096	0.7088

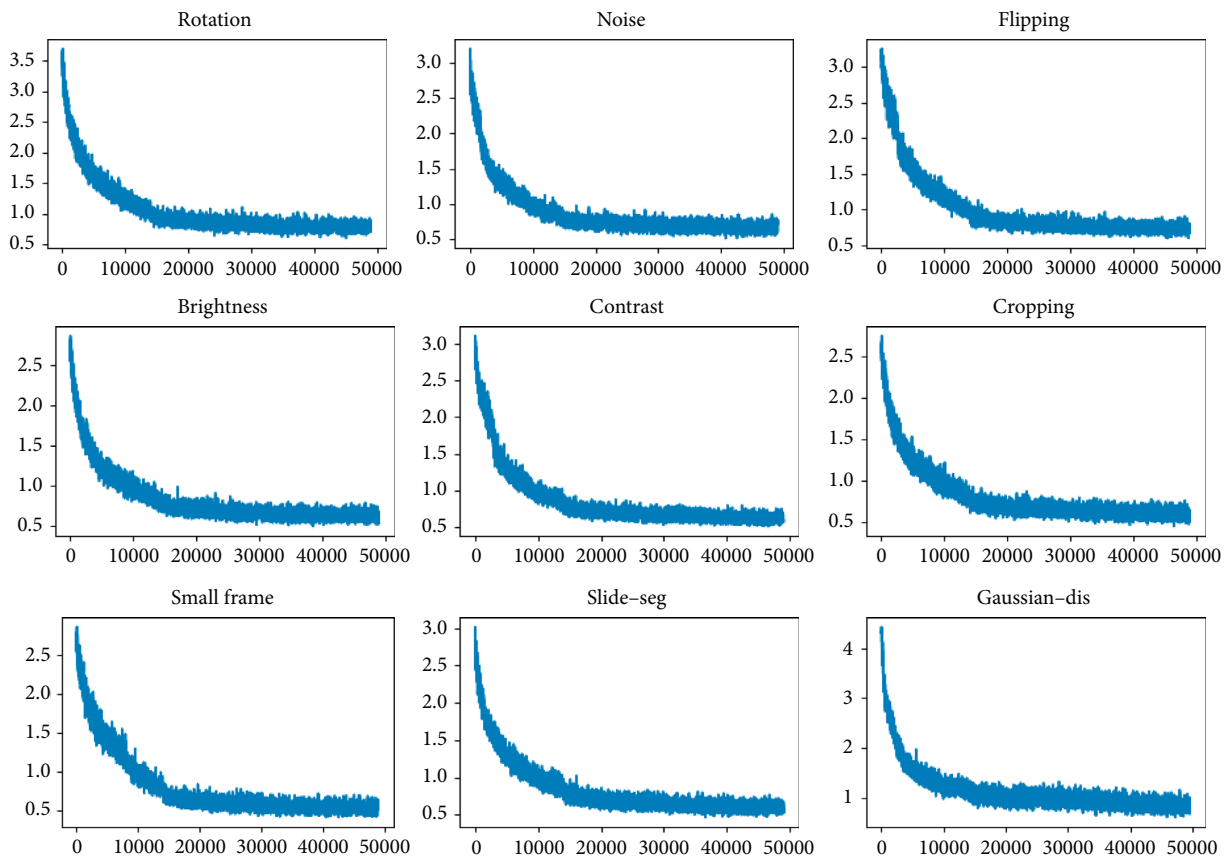


Figure 6: The loss changes of all methods in the training process on YOLO. We can see that the loss value of all methods will eventually converge to less than 1.

It can be observed that no matter what kind of augmentation method is used, there is a step-by-step loss drop between the 10000 and 20000 rounds of iterations, which proves that our learning rate decay strategy is effective under these methods. We have counted the best test results of each model in the above experiments, and these data are helpful for us to judge which methods are helpful for the test of vehicle logo data. The results are as in Table 2.

The experimental results show that the cropping operation improves the results the most in the traditional method, and the $F1$ value increases by 2.63 percentage points. Compared with the original dataset, the $F1$ values of the three methods proposed in this paper increased by 4.11, 4.04, and 5.56 percentage points, respectively. The results show that our method is more effective than the traditional method to help the network to detect the vehicle logo.

We choose the best three of the traditional methods and our proposed method for further experiments to observe the effect of our proposed method in combination. In order to ensure the accuracy of the conclusion, we test the data in the framework of YOLO and Faster R-CNN, respectively, and get the results of $F1$ value in Figure 7.

It can be seen that, compared with the traditional methods, our methods can improve the performance better under the framework of YOLO and Faster R-CNN. We select the best test results of various methods and make them into Table 3.

Our method mainly optimizes the scale diversity and location diversity of the target. Based on the characteristics of large differences between classes and small differences within classes, we propose a new data optimization method. Compared with the traditional method, we mainly focus on two points with the least amount of information in the small dataset of vehicle logo. Experiments show that our idea is effective.

From the experimental results, we can know that the improvement effect of our method on the YOLO framework is better than that of Faster R-CNN. Our method is 3.77 percentage points higher than the traditional method in the

Table 2: The best test results of all methods on YOLO.

	Rotate-on	Noise	Flipping	Brightness	Contrast	Cropping	Small frame	Slide-seg	Gaussian-dis
P	0.8539	0.8521	0.8492	0.8422	0.8641	0.8577	0.9139	0.8862	**0.9151**
R	0.6077	0.6096	0.6170	0.6395	0.6311	0.6433	0.6358	0.6489	**0.6564**
F	0.7101	0.7107	0.7147	0.7270	0.7294	0.7351	0.7499	0.7492	**0.7644**

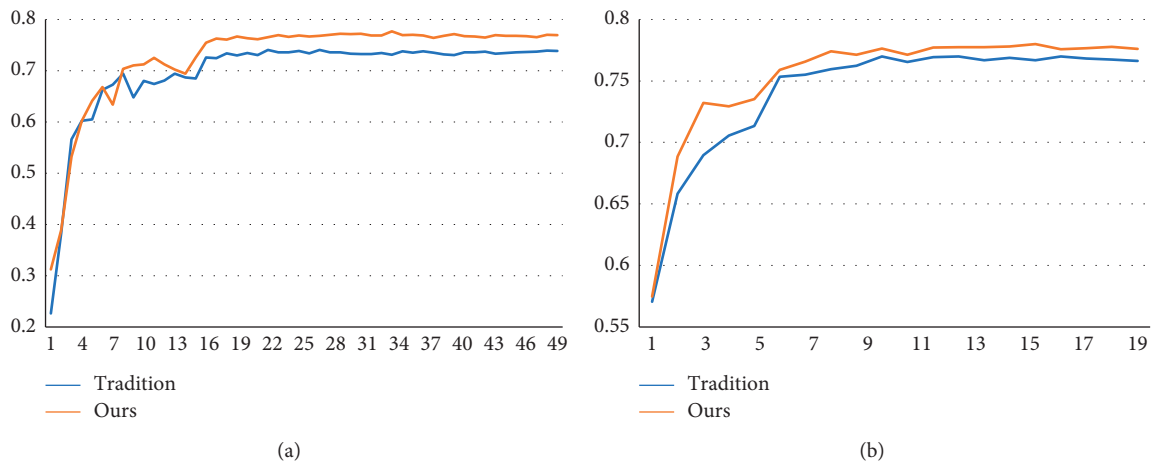


Figure 7: Results of controlled experiments. Although the traditional method sometimes works better than our method in the early stage of training, the final result is that our method is dominant. (a) Result curves of YOLO. (b) Result curves of Faster R-CNN.

Table 3: The best detection results of two networks in each dataset.

	YOLO			Faster R-CNN		
	Origin	Tradition	Ours	Origin	Tradition	Ours
P	0.8466	0.8625	**0.9295**	0.8885	0.9127	**0.9358**
R	0.6096	0.6461	**0.6667**	0.4775	0.6657	**0.6685**
*F*1	0.7088	0.7388	**0.7765**	0.6212	0.7699	**0.7799**

Table 4: Some promotion results show.

	YOLO	Faster R-CNN
Origin		
Tradition		
Ours		

framework of YOLO, which is better than Faster R-CNN. The reason is that we mainly optimize the size and location of the vehicle logo, which is more targeted for the one-stage method like YOLO. Due to the existence of RPN network, Faster R-CNN is not very sensitive to the location of the target, and $F1$ only increases by 1.0 percentage points. As a one-stage method, the YOLO framework can directly return the location of the target from the image. Previous studies have pointed out that CNN is more sensitive to the location of the target in the image, so the regression of the target location in the YOLO framework is more dependent on the dataset. Our method is mainly optimized for the diversification of logo location, so our method performs better under the framework of YOLO. Therefore, although the method proposed in this paper can improve the performance of both networks, it is more obvious to improve the performance of YOLO.

4.3. Results Show. We show some detection effects after data enhancement in Table 4.

In Table 4, the first column of each group of images shows the help of our proposed method for improving the network recall rate. For the image in the first column, the network trained by the original dataset and the dataset enhanced by the traditional method cannot detect the vehicle logo in the image, and the data enhancement method proposed by us can successfully detect the vehicle logo. The second column of each group of images shows the improvement of accuracy of our method. For the error detection in the network trained by the original dataset and the dataset enhanced by the traditional data method, our method can suppress it to a certain extent.

5. Conclusion

In this paper, we propose a series of data optimization methods for small sample vehicle logo dataset. According to the characteristics of small sample vehicle logo dataset, we enhance the dataset from two aspects: size and location. The experimental results show that the methods proposed in this paper can improve the recall and precision compared with the traditional method, whether alone or in combination. For different frameworks, our method is more suitable for the one-stage methods that directly find the target position from the image, so the improvement effect of Faster R-CNN is weaker than that of YOLO.

Data Availability

The image data used to support the findings of this study have been deposited in the Baidu Netdisk repository (https://pan.baidu.com/s/10LG6vZGK_tEV6sGbDx4uug).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grant nos. 61972097, 61502105, and 61672158, the Technology Guidance Project of Fujian Province under Grant no. 2017H0015, the Natural Science Foundation of Fujian Province under Grant no. 2018J1798, the Fujian Natural Science Funds for Distinguished Young Scholar under Grant no.2015J06014, the University Production Project of Fujian Province under Grant no. 2017H6008, the Fujian Collaborative Innovation Center for Big Data Application in Governments, and the Fujian Engineering Research Center of Big Data Analysis and Processing.

References

[1] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-enabled intelligent transportation systems for the smart city: applications and challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.

[2] B. Chen, L. Shi, and Ke. Xiao, "A robust moving object detection in multi-scenario big data for video surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 982–995, 2018.

[3] T. Lai, H. Wang, Y. Yan, T.-J. Chin, and W.-L. Zhao, "Motion segmentation via a sparsity constraint," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 973–983, 2017.

[4] Y. Niu, W. Lin, and X. Ke, "CF-based optimisation for saliency detection," *IET Computer Vision*, vol. 12, no. 4, pp. 365–376, 2018.

[5] X. Ke and W. Guo, "Multi-scale salient region and relevant visual keywords based model for automatic image annotation," *Multimedia Tools and Applications*, vol. 75, no. 20, pp. 12477–12498, 2016.

[6] Ke Xiao, L. Shi, W. Guo, and D. Chen, "Multi-Dimensional traffic congestion detection based on fusion of visual features and convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2157–2170, 2019.

[7] İ. Türkyılmaz and K. Kaçan, "License plate recognition system using artificial neural networks," *ETRI Journal*, vol. 39, no. 2, pp. 163–172, 2016.

[8] Y. Yuan, W. Zou, Z. Yong et al., "A robust and efficient approach to license plate detection," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1102–1114, 2017.

[9] S. Yu, S. Zheng, H. Yang et al., "Vehicle logo recognition based on bag-of-words," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 353–358, IEEE, Krakow, Poland, August 2013.

[10] B. Cyganek and M. Woźniak, "An improved vehicle logo recognition using a classifier ensemble based on pattern tensor representation and decomposition," *New Generation Computing*, vol. 33, no. 4, pp. 389–408, 2015.

[11] D. F. Llorca, R. Arroyo, and M. A. Sotelo, "Vehicle logo recognition in traffic images using HOG features and SVM," in *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems*, pp. 2229–2234, IEEE, Hague, Netherlands, October 2013.

[12] H. Peng, X. Wang, H. Wang et al., "Recognition of low-resolution logos in vehicle images based on statistical random

sparse distribution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 681–691, 2015.

[13] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding, "Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1951–1960, 2015.

[14] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, "Vehicle logo recognition using whitening transformation and deep learning," *Signal, Image and Video Processing*, vol. 13, no. 1, pp. 111–119, 2019.

[15] A. P. Psyllos, C. N. E. Anagnostopoulos, and E. Kayafas, "Vehicle logo recognition using a SIFT-based enhanced matching scheme," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, 2010.

[16] Y. Liu and S. Li, "A vehicle-logo location approach based on edge detection and projection," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*, pp. 165–168, IEEE, Beijing, China, July 2011.

[17] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, "Hyper-parameters optimisation of deep CNN architecture for vehicle logo recognition," *Iet Intelligent Transport Systems*, vol. 12, no. 8, pp. 939–946, 2018.

[18] J. Redmon, S. Divvala, R. Girshick et al., "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, IEEE, Las Vegas, NV, USA, June 2016.

[19] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society,Honolulu, HI, USA, July 2017.

[20] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[21] R. Girshick, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Columbus, OH, USA, June 2014.

[22] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Santiago, Chile, December 2015.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[26] A. Neubeck and L. V. Gool, "Efficient non-maximum suppression," in *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 850–855, IEEE, Hong Kong, China, August 2006.

[27] H. Li and C. Y. Suen, "Robust face recognition based on dynamic rank representation," *Pattern Recognition*, vol. 60, no. 12, pp. 13–24, 2016.

[28] S. K. Biswas and P. Milanfar, "Linear support tensor machine with LSK channels: pedestrian detection in thermal infrared images," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4229–4242, 2017.

[29] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, pp. 146–157, 2018.

[30] X. Ke, M. Zhou, Y. Niu, and W. Guo, "Data equilibrium based automatic image annotation by fusing deep model and semantic propagation," *Pattern Recognition*, vol. 71, pp. 60–77, 2017.

[31] R. Liu and D. F. Gillies, "Overfitting in linear feature extraction for classification of high-dimensional image data," *Pattern Recognition*, vol. 53, pp. 73–86, 2016.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 5605479, 10 pages
https://doi.org/10.1155/2020/5605479

Research Article

Demand Analysis and Distribution of Single-Trip Ticket Cards for Urban Rail Transit

Zi-jia Wang,[1,2] **Jing-qi Li**,[1] **Jiang-yue Wu**,[3] **and Zhi-gang Yang**[4]

[1]*Department of Civil Engineering, Beijing Jiaotong University, Beijing 100044, China*
[2]*Beijing Engineering and Technology Research Center of Rail Transit Line Safety and Disaster Prevention, No. 3 Shangyuancun, Haidian District, Beijing 100044, China*
[3]*Department of Urban Planning and Design, The University of Hong Kong, Hong Kong 999077, China*
[4]*The Traffic Planning Department, Beijing Municipal Institute of City Planning & Design (BICP), No. 60 Nanlishi Street, Beijing 100045, China*

Correspondence should be addressed to Jiang-yue Wu; wujiangyue@hotmail.com

Received 21 November 2019; Revised 21 February 2020; Accepted 13 April 2020; Published 7 July 2020

Guest Editor: Feng-Jang Hwang

In the current urban rail transit systems, nearly 15% of passengers are noncommuter travelers who use single-trip ticket cards (ticket cards). Accordingly, the effective management of ticket cards is of great importance. This article suggests a time series model for use in predicting ticket card storage based on the characteristics of ticket cards collected by an automatic fare collection (AFC) system. The distribution cycle, station types, and distribution volume of each station are also determined. Then, drawing on small package transportation feasibility theory, an unbalanced distribution model between production and demand (unbalanced distribution model), as well as a hybrid distribution model of loading and unloading (hybrid distribution model), is established. Application of these models to the Beijing Subway system is used to verify the efficiency and feasibility of such a hybrid distribution model. The analysis and results offer insights into usage patterns of urban rail transit ticket cards, providing solid evidence for a relative decision-making process.

1. Introduction

One notable barrier to effective management of ticket cards is the geographical imbalance of ticket card storage among subway system's stations. After ticket cards are sold by an AFC system, they are collected by an automatic gate machine (AGM) as passengers exit the station. However, uncertain passenger flows inevitably result in geographical imbalances of ticket cards. Furthermore, ticket storage capacity management in each station is based on the assumption that "more is better." Flexible distribution of ticket cards based on the specific circumstances of each station has thus far rarely been investigated. However, the present approach to managing ticket cards is not sufficient to effectively respond to growing traffic demand. In response to these issues, this paper proposes a set of positive management methods for forecasting and dynamically distributing ticket cards among stations so as to reduce total storage demand for ticket cards, improve ticket card turnover rate, and ensure better travel experiences for passengers.

Precise short-term predictions of passenger flows are the basis for determining distribution cycle, distribution volume, and other basic statistics of ticket cards. Many existing studies have explored short-term passenger flow prediction. After Box and Jenkins [1] proposed a systematic method for identifying, analyzing, and predicting time series, a number of other researchers improved their time series model [2, 3], applying it more broadly [4]. In recent years, the emergence of big data has done much to improve the stability and applicability of the time series model. As the data collected become complicated, many data mining methods are used as a complement operator to the classic ARIMA model, processing unstable smart-card data. The hybrid model is also used to predict short-term passenger flows. Common

applications include the ARIMA-Kalman filter model [5], ARIMA-ANN model [6], ARIMA-SVM model [7], ARIMA-GA model [8], and EMD-BPN [9] model. Thus, the time series model has been proved to be an effective approach with which to predict short-term passenger flows in an urban transit system. This paper uses the classical time series model for prediction, based on AFC records, to establish a storage prediction model.

Second, although ticket card distribution is seldom addressed in the literature, distribution itself has been studied extensively, especially in relation to surface transportation for time and cost savings. Many logistics models have been introduced to solve problems relating to delivery of goods. However, little research has been conducted on dynamic distribution in a subway setting primarily because single-trip ticket cards are rarely recycled and because the potential cost savings associated with distribution models have been underestimated by decision makers and subway companies alike. To tackle the distribution problem in a subway environment, a logistics model is needed that reflects specific constraints corresponding to the actual situation of each station. This paper mainly introduces two distribution models, the unbalanced distribution model and the hybrid distribution model, which are based on two logistic models: the unbalanced production-marketing problem and the vehicle routing problem with simultaneous pickup and delivery (VRPSDP). The hybrid distribution model offers more cost savings than the unbalanced distribution model by considering both "pickup" and "delivery" demands simultaneously, thus saving time and manpower. Within the context of card collection and distribution, "pickup" refers to the collection of surplus cards at a station and "delivery" to the transporting of surplus cards to stations having a card deficit. Essentially, it can be considered an extension of VRPSDP.

Clarke and Wright [10] proposed the traveling salesman problem (TSP), a special case of the VRPSDP. Min [11] first abstracted the VRPSDP model from the real-world problem of delivering books among libraries and solved it using the genetic algorithm. Since then, the VRPSDP model has been gradually improved to apply to different practices, including by adding time windows [12], allowing multiple vehicles [13], and reservations for repeated service [14]. Beyond improvements to constraints, its objective is becoming increasingly generalized from just distance [14] and time [12] to generalized cost savings, such as through manpower arrangements for accomplishing distribution [13]. At present, methods of solving the VRPSDP include mainly stereotype algorithms [15] and heuristic algorithms. Instead of providing the best solution, heuristic algorithms are more adaptive to big data demands—for example, tabu research [16], the tourism partition heuristic algorithm [17], the genetic algorithm [18], simulated annealing [19], and ant colony systems [20]. To improve algorithmic efficiency and avoid single method limitations, a hybrid algorithm is widely accepted as tabu research with guided local research [21], ant colony systems with a tabu research operator [22], and a combination of the metaheuristic algorithm [23], and the like, and has become a focus of mainstream research. The genetic algorithm is particularly powerful when used in the VRPSDP logistics model. It has been used to tackle certain types of the vehicle routing problem in recent decades, often in combination with other algorithms to improve its effectiveness. However, existing studies have mostly concentrated on applications to road traffic problems, with few having focused on distribution balance. This paper proposes a distribution model based on the urban rail transit network and the characteristics of ticket card distribution.

This paper is organized as follows: Section 2 proposes a storage prediction model based on a time series model. Section 3 develops a reasonable distribution model on the basis of the forecast data. Section 4 offers a feasible distribution plan using data for the Beijing Subway from December 2016.

2. Storage Prediction Model of Single-Trip Ticket Cards

A time series model is used to predict ticket card data. For this step, ticket card flow volume, distribution volume, and storage settings are considered, which are defined in the current management standards [24] as follows:

(1) Flow volume:

$$F = S - R, \tag{1}$$

where F is the flow volume of ticket cards, S is the total volume of ticket cards sold, and R is the recycled number of ticket cards.

(2) Distribution volume:

The distribution volume of ticket cards is the number of ticket cards to be deployed from stations that have a ticket card surplus to those that have a deficit.

(3) Storage capability settings:

$$C_{\max} = a * \overline{S} * 120\%, \tag{2}$$

$$C_s = a * \overline{S}, \tag{3}$$

$$C_{\min} = a * \overline{S} * 80\%, \tag{4}$$

where $C_{\max}$ is the maximum storage capability, C_s is the safe storage capability, $C_{\min}$ is the minimum storage capability, a is a coefficient that equals 5 for public holidays but equals 4, otherwise, and $\overline{S}$ is the average daily sales volume of the last distribution cycle.

2.1. ARIMA Model. ARIMA(p, d, q) is the most general form of the classical time series model. The classical ARIMA model can be applied only to stable time series, so the test ADF is used to judge whether a time series is stable. Then, parameters can be estimated, and the t-test and q-test are used to ensure that the ARIMA model is error-tolerant compared with the actual times series. Several possible ARIMA models may be proposed in the foregoing steps, after which the AIC test [25] is used to select the best ARIMA

model. Using the inverse difference operator, final prediction results are acquired. The general process of establishing the ARIMA(p, d, q) model [26–28] is shown in Figure 1.

According to the general process of the ARIMA(p, d, q) model, this paper uses STATA [29] to make a short-term prediction concerning the sales volume and flow volume of each station. Data cover the period from December 19, 2016, to December 25, 2016.

2.2. Prediction Model. Depending on ticket card characteristics, the minimum storage capacity of each station should satisfy the sales volume of the first day in this distribution cycle as well as in the next. The associated mathematical expressions are given in equations (5) and (6):

$$C_{\min} > S_1, \tag{5}$$

$$C_{\min} - \sum_{i=1}^{N} F_i > S_{N+1}, \tag{6}$$

where S_i is the sales volume or the prediction of sales volume, $i = 1, 2, \ldots, N + 1$, F_i is the flow volume or prediction of flow volume, $i = 1, 2, \ldots, N$, and N is the prediction cycle.

The ARIMA model is applied to each station to estimate its sales volume for the first day in the next distribution cycle. Using equations (5) and (6), final minimum storage capability is obtained. The number of ticket cards in one ticket box is added to the final minimum storage capability of 2,000 pieces to address uncertainty about passenger flow. Under the current regulations (equations (2) and (4)), the relative ratio between the maximum storage capability and the minimum storage capability is fixed, so the maximum storage capability can be determined.

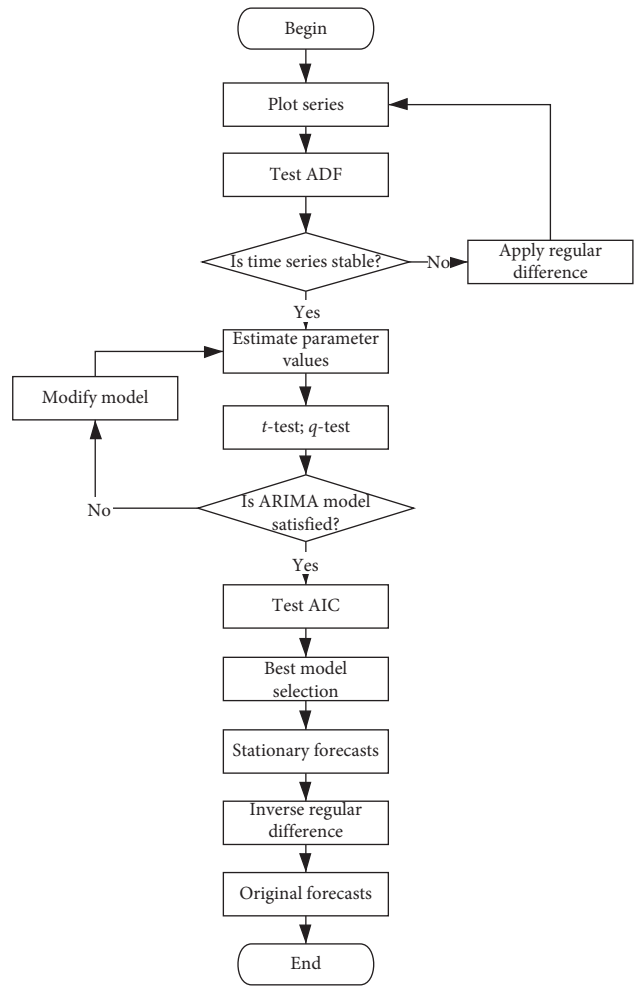


Figure 1: Classical time series model flowchart.

3. Balanced Distribution Model of Single-Trip Ticket Cards

The subway itself is believed to be an effective carrier for distributing ticket cards. A large number of existing studies have demonstrated the rationality and feasibility of using the subway to transfer goods during nonpeak hours. Liu [30] and Shen [31] studied the feasibility of using the Beijing Subway system as a nighttime logistics circulation system in urban areas. Some studies [32, 33] have also explored the feasibility of using the Tyne rail transit for cargo transport in the United Kingdom. The benefits of using rail transit as a carrier include improving the utilization rate of the subway in low-peak hours and avoidance of technical problems such as alterations to subway coaches and subway tracks. Ticket cards are smaller and easier to transport than other heavy goods, confirming the feasibility of distributing ticket cards using the subway system itself.

In distribution models, station-to-station distribution is often applied to actual situations, especially distribution between an AFC clearing center (ACC) and subway stations. Furthermore, in practice, distribution follows a "more is better" approach, so subway companies generally neglect full utilization of ticket cards and manpower. An unbalanced distribution model is established as a simulation of the actual distribution, and a hybrid distribution model is proposed as an effective alternative. In this part, some necessary parameters and concepts used in distribution models are defined, after which two distribution models are introduced. Next, the reasons for proposing these two models, with a particular focus on total distance and management cost, are illustrated in detail.

3.1. Distribution Volume. Assuming that the current storage of a station is c, the number of ticket cards to be allocated is

$$\Delta c = C_s - c, \tag{7}$$

where Δc is the number of ticket cards to be distributed, C_s is the safe storage capability, and c is the current storage.

For a station x,

$$\begin{cases} \Delta c > 0, & \text{station } x \text{ is a loss station,} \\ \Delta c < 0, & \text{station } x \text{ is a surplus station.} \end{cases} \tag{8}$$

Whether the station participates in the distribution depends on distribution volume and actual sales volume.

Different types of stations should have different standards. For a loss station i,

$$\begin{cases} |\Delta c| > S_l, & \text{station } i \text{ participates in the deployment,} \\ |\Delta c| < S_l, & \text{otherwise,} \end{cases} \tag{9}$$

where S_l is the loss number threshold of ticket cards.

Similarly, for a surplus station j,

$$\begin{cases} |\Delta c| > S_s, & \text{station } j \text{ participates in the deployment,} \\ |\Delta c| < S_s, & \text{otherwise,} \end{cases} \tag{10}$$

where S_s is the surplus number threshold of ticket cards.

3.2. Distribution Model Unbalanced between Production and Demand. In practice, total distribution between surplus stations and loss stations is not equal, according to Beijing AFC data. The ACC conducts the distribution of the subway system and provides or recycles ticket cards. Suppose that the total surplus is greater than the total deficit—specifically that m stations have a surplus of ticket cards and n stations a deficit. Because more ticket cards need to be reused, ACC is taken as a station with a deficit of ticket cards so that the total number of loss stations is $n+1$. Based on the unbalanced logistics model, some constraints in the context of the subway system are added, with the ultimate mathematical expressions of the unbalanced distribution model given in equations (11) to (14):

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n+1} D_{ij} x_{ij}, \tag{11}$$

$$\text{s. t} \sum_{j=1}^{n+1} x_{ij} = s_i, \quad i = 1, 2, \ldots, m, \tag{12}$$

$$\sum_{i=1}^{m} x_{ij} = l_j, \quad j = 1, 2, \ldots, n+1, \tag{13}$$

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n+1} l_j, \tag{14}$$

where x_{ij} is the distribution volume between surplus station i and loss station j ($x_{ij} \geq 0$), D_{ij} is the distance between surplus station i and loss station j, s_i is the total distribution volume of surplus station i ($s_i > 0$), l_j is the total distribution volume of loss station j ($l_j > 0$), m is the number of surplus stations, and n is the number of loss stations.

3.3. Hybrid Distribution Model of Loading and Unloading. According to the logistical model that is the basis of the unbalanced distribution model, the foregoing distribution model determines optimal station-to-station distributions. According to the Ticket Distribution Management Regulations [34], at least one ticket management crew and one security guard are required for each station-to-station distribution. Accordingly, this method will cause significant wastage of human resources. Furthermore, distribution itself is arranged at night or other nonpeak hours when in response to reduced workload, fewer staff are present. As a result, human resources should be more fully utilized. Another disadvantage of unbalanced distribution is that no distribution tasks are involved in the backhaul of station-to-station distribution, meaning that the objective value, the total distance, can still be reduced by a change in distribution methods.

Based on the two preceding reasons, the VRPSDP is used to establish a hybrid distribution model. Assuming that n stations participate in distribution, the total surplus exceeds the total loss. ACC is equivalent to a loss station, and its station number is 0. Thus, the hybrid distribution model can be expressed as

$$\min \sum_{i=0}^{n} \sum_{j=0}^{n} D_{ij} x_{ij}, \tag{15}$$

$$\text{s.t} \sum_{i=0}^{n} x_{ij} = 1, \quad i \neq j, \tag{16}$$

$$\sum_{j=0}^{n} x_{ij} = 1, \quad i \neq j, \tag{17}$$

$$z_{ij} \geq 0, \quad i \neq j, \tag{18}$$

$$\sum_{j=1}^{n} z_{0j} = \sum_{i=1}^{n} (l_i - s_i), \tag{19}$$

$$\text{or} \sum_{j=1}^{n} z_{0j} = \sum_{i=1}^{n} (s_i - l_i), \tag{20}$$

$$\sum_{i=0}^{n} z_{ij} + (s_i - l_i) \sum_{i=0}^{n} x_{ij} = \sum_{i=0}^{n} z_{ji+1}, \tag{21}$$

$$\begin{aligned} l_i &> 0, \\ \text{or } s_i &> 0, \end{aligned} \tag{22}$$

$$x_{ij} = \begin{cases} 1, & \text{there is deployment between station } i \text{ and station } j \\ 0, & \text{otherwise} \end{cases} \quad (i \neq j), \tag{23}$$

where D_{ij} is the distance between station i and station j, $i \neq j$, x_{ij} is a parameter, $i \neq j$, z_{ij} is the number of ticket cards involved in distribution from station i to station j, l_i is the number of ticket cards to bring to station i, where if $l_i > 0$, then station i is a loss station, and s_i is the number of ticket cards to bring out for station i, where if $s_i > 0$, then station i is a surplus station.

Objective function (15) minimizes the total distance. Constraints (16) and (17) mean that each station can be served only once; constraint (18) ensures that the number of ticket cards during the distribution is positive; and

Table 1: Prediction data of Wangfujing Station.

Date	Dec. 23, 2016	Dec. 24, 2016	Dec. 25, 2016	Dec. 26, 2016
Flow volume(pieces)	1,242	1,618	1,767	
Sales volume (pieces)	9,593	14,421	12,690	14,224

constraint (19) indicates that when starting from ACC, the number of ticket cards equals the sum of the distribution number in all loss stations minus the sum of the distribution number in all surplus stations. Constraint (20) represents the opposite situation. Only one of these two constraints can be met in the actual circumstances. Constraint (21) ensures that the demand of ticket cards for each station could be met; constraint (22) requires that the distribution volume be positive.

4. Case Study

To better investigate the performance of the preceding models, the Beijing Subway is taken as a real-world example. In the Beijing Subway system, single-trip ticket cards account for 15% of all total ticket cards in the AFC system. The data used in the case study cover the period from December 23, 2016, to December 26, 2016. We took the initial number of ticket cards for all stations to be baseline of safe storage capability on the beginning of December 23. Section 4.1 analyzes the advantages and feasibility of the storage prediction model and then determines the distribution cycle of ticket cards; Section 4.2 solves, compares, and analyzes the two distribution models in a MATLAB environment.

4.1. Distribution Cycle. Taking Wangfujing Station as an example, we used the prediction model to forecast the associated statistics, given in Table 1.

Wangfujing Station is a loss station, with an average sales volume of 12,732 pieces. Under the current regulations, the storage for the next distribution cycle should be 40,743–61,114 pieces. However, a storage predicting model predicts that storage will be 20,851–31,277 pieces or 49% less.

Thus, the storage prediction model can significantly reduce total storage of ticket cards while improving ticket card circulation frequency and utilization rate. However, because the model has some limitations that prevent it from guaranteeing an opportune response to unexpected fluctuations in passenger flow, storage settings should be raised commensurately in actual application. Even with such adjustments, the Wangfujing example indicates that the storage predicting model can still save ticket card resources and attendant costs of management.

Determination of the distribution cycle is intended to stabilize the number of stations involved and the total distribution volume of ticket cards within a reasonable range. According to relevant regulations [24] and studies [35] of the Beijing Subway, S_l is set at 2,000 and S_s at 1,000. The number of stations to be deployed after day's end is shown in Figures 2 and 3.

Figure 2 shows a steep increase on the fifth day in both number of loss stations and number of surplus stations,

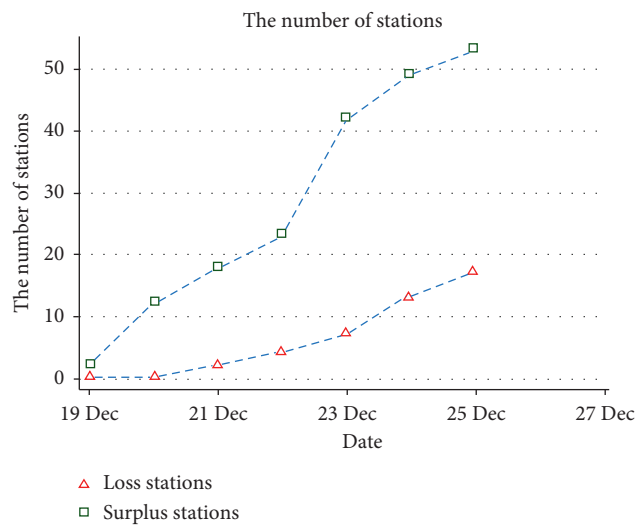


Figure 2: The number of stations to be deployed 1.

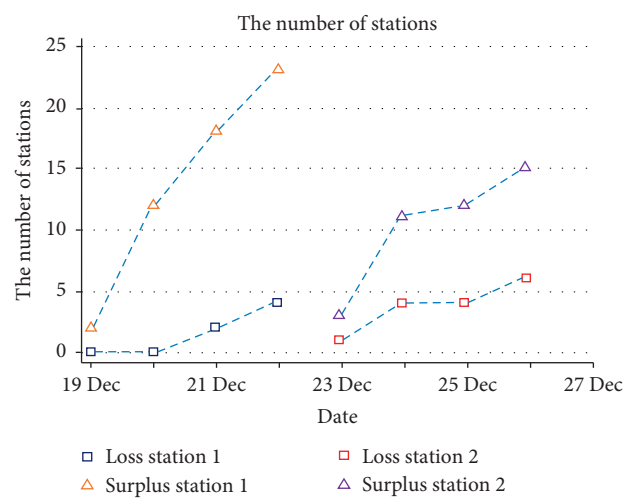


Figure 3: The number of stations to be deployed 2.

indicating that most stations need distribution after 4 days' operation. If setting the distribution cycle to a week, total distribution volume will increase to 55,129, and the total number of stations to be deployed to about 70, or fourth of total stations—a difficult task to accomplish in a single distribution. Accordingly, 4 days is a more suitable threshold than a week. For a 4-day distribution cycle, as Figure 3 shows, the total number of stations that needs to participate in distribution is approximately 25, of which about 5 are loss stations. Not only the number of distributed stations but also total distribution volume varies within an acceptable range, with 19,780 ticket cards needing deployment in the first 4 days and 21,249 in the next 4. Accordingly, the distribution

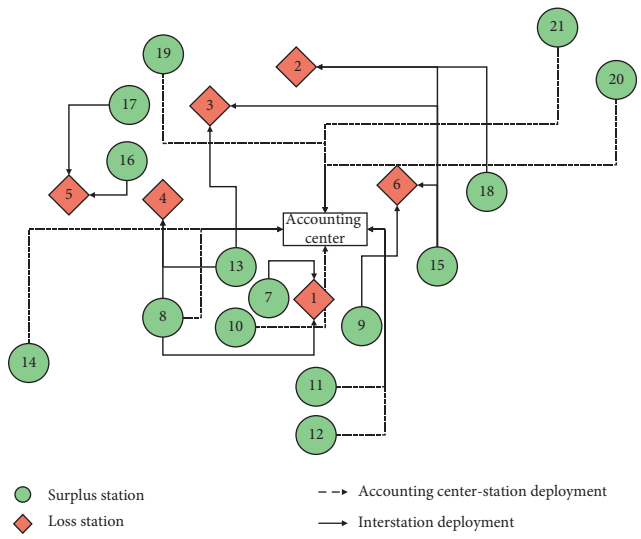


Figure 4: Schematic of the unbalanced distribution model.

Table 2: Partial simulation results.

Number	Total iterations	Shortest path	Number of best iterations	Total distance (km)
1	100	0–14–8–1–7–10–12–11–9–6–18–3–2–0; 0–21–20–15–13–4–5–16–17–0; 0–19–0	70	215.36
2	150	0–21–20–15–6–10–9–13–4–5–16–17–0; 0–14–8–1–7–12–11–18–3–2–0; 0–19–0	66	235.75
3	150	0–8–7–9–10–12–11–13–4–6–18–3–2–0; 0–21–20–15–1–14–5–16–17–0; 0–19–0	106	227.4
4	200	0–14–8–11–12–10–7–1–9–6–18–2–0; 0–21–20–15–13–4–3–5–16–17–0; 0–19–0	132	220.05
5	200	0–14–8–7–1–9–11–12–10–6–18–2–0; 0–21–20–15–13–4–3–5–16–17–0; 0–19–0	121	218.88
6	200	0–21–20–15–6–18–3–2–0; 0–14–8–1–11–12–9–10–7–13–4–5–16–17–0; 0–19–0	92	215.35

cycle for nonholidays is determined to be 4 days. Because holidays in China generally last 3 days, the distribution cycle for holidays is set to 3 days.

4.2. Distribution Plan. According to the foregoing analysis, 21 stations will need distribution on December 26, 2016: 6 loss stations and 15 surplus stations. ACC participates in the distribution as well.

In two distribution models, the Dijkstra algorithm [36] is used to solve for key parameter D_{ij}, the shortest distance between two stations. ACC is a certain location at the center of all stations, so D_{0j} or D_{i0} is simplified as the mean of all distances between ACC and each station: 17 km. Results show that total distance in the unbalanced distribution model is 550.5 km, with 21 station-to-station distributions. Figure 4 shows a schematic of the distribution.

A few points should be explained concerning the hybrid distribution model:

(1) When calculating fitness, two constraints were added to improve distribution reasonableness. The first is intended to limit the degree of dispersion among stations. If the cumulative distance reaches 100 km (a value determined through multiple experiments), the crew should return to the ACC and restart distribution with a new branch line. The second is that the remaining number of ticket cards must meet the demand of the next station in the branch line after serving one station. If it does not, the crew should return to the ACC and restart with a new branch line.

(2) The population size is set at 400 and the number of iterations at 200 (values determined through

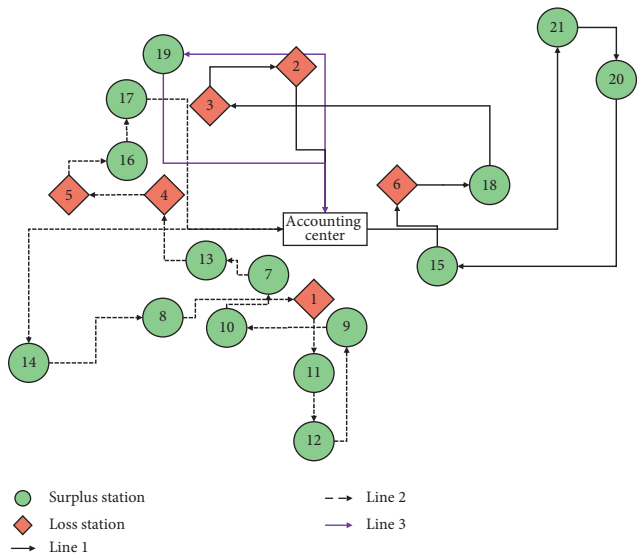


Figure 5: Schematic of the hybrid distribution model.

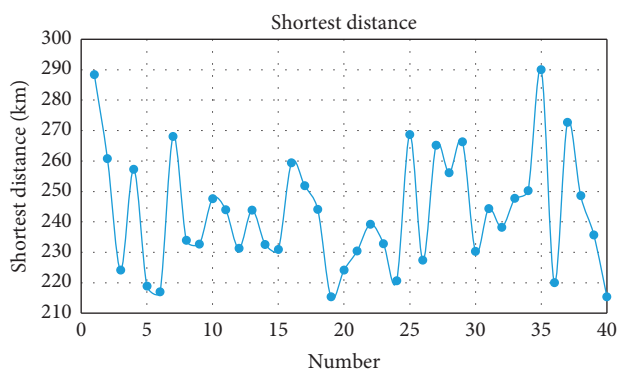


Figure 6: Schematic of the optimal experimental convergence result.

multiple experiments), with a probability of chromosome crossing PC (term of genetic algorithm) of 0.85 and a probability of variation PM of 0.01.

To obtain better optimization results, several simulation experiments were carried out. Table 2 shows the results of partial experiments.

The best experimental result is for 215.35 km, with three branch lines. Figure 5 shows a schematic of distribution.

Figure 6 shows the convergence result of the optimal experiment.

Clearly, the final convergence result is inconsistent with the optimal solution, which appears in the early stage. Considering the change of path, it can be reasonably inferred that the final convergence result is generated from the optimal solution, which is itself generated by the cross-over operation—indicating that room remains for improvement of the algorithm.

To solve this problem, a simulated annealing operator [37] is added to enhance the global search ability and prevent the optimal chromosome from occupying the entire population too quickly. We set the initial temperature t_0 to 90 and the attenuation coefficient α to 0.95. Because simulated annealing reduces convergence speed, the number of iterations is doubled to 400. Table 3 gives partial experimental results.

Figure 7 shows a significantly better convergence result than can be achieved using a single algorithm, albeit a slight difference in the minimum distance. In addition, the total number of lines is not limited to three, indicating that the diversity of itineraries is also increased. These results thus demonstrate the greater strength of the algorithm's global search ability.

In conclusion, the hybrid distribution model significantly outperforms the unbalanced distribution model. Not only is the total distance clearly reduced but distribution efficiency is also greatly increased. In addition, the unbalanced distribution plan involves 21 station-to-station distributions, requiring at least 21 groups of staff, whereas in the hybrid distribution model, three groups of personnel can be assigned to the independent lines. Freed from any time limit, one group of personnel would be sufficient, greatly improving the utilization rate of human resources. The genetic

Table 3: Partial simulation results.

Number	Total iterations	Shortest path	Number of best iterations	Total distance (km)
1	400	0–19–17–16–5–6–9–10–11–12–1–7–0; 0–21–20–15–18–2–3–13–4–8–0; 0–14–0	304	250.35
2	400	0–20–21–18–6–5–16–3–2–0; 0–13–4–11–12–10–7–1–9–15–8–14–17–0; 0–19–0	321	246.05
3	400	0–21–20–4–13–12–11–5–16–17–0; 0–14–10–7–8–1–9–15–6–18–3–2–0; 0–19–0	292	232.7
4	400	0–19–17–16–5–14–0; 0–21–0; 0–4–0; 0–13–12–11–10–7–8–1–9–15–6–18–3–2–0; 0–20–0	383	235.7
5	400	0–10–9–11–12–6–15–8–7–1–13–4–5–0; 0–19–17–16–3–18–2–0;0–21–20–0; 0–14–0	311	253.18
6	400	0–21–20–15–6–18–3–2–0; 0–14–8–1–7–10–9–11–12–13–4–5–16–17–0; 0–19–0	361	213.5

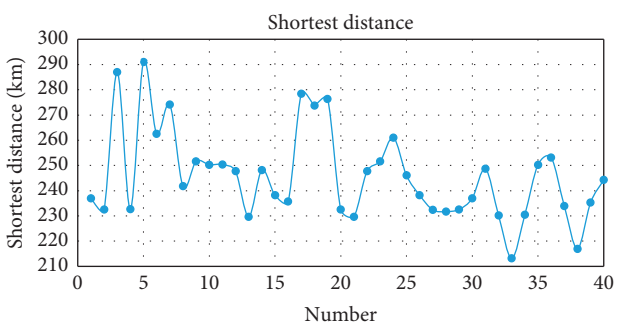


Figure 7: Schematic of the optimal experimental convergence result (with the simulated annealing operator).

algorithm used in the hybrid distribution model is highly adaptable to various practical problems. Accordingly, after a comprehensive comparison, the hybrid distribution model offers better simulation effect, much room for improvement, and a more convincing practical application.

5. Conclusions

This paper, being based on AFC data, mainly studies forecast and distribution problems related to ticket cards. The current distribution follows the principle of "more is better," having the problem of low recycled rate and ineffective manpower arrangements. To propose a feasible and efficient distribution model, a forecast model of sales volume and a determination of recycled threshold are demanded. With the prediction of loss stations, surplus stations, and the total distribution volume of ticket cards, a hybrid distribution model is introduced and compared with the current distribution model. Its primary conclusions are as follows:

(1) A quantitative method extended from times series prediction for determination of storage settings is established that can improve empirical determinations. Based on a case study, the storage predicting model can reduce total storage 40%, on average, from that seen under the current regulations.

(2) By employing the prediction model, distribution value, loss stations, and surplus stations are determined. The distribution cycle is set as 4 days for nonholidays and 3 days, otherwise, on the basis of ticket card data.

(3) A new distribution model is established using the theory of reverse logistics. Experimental results show that searching behavior can be enriched, with convergence and diversity stressed and well balanced with multiple operators. Compared with the current distribution method, simulations demonstrate the effectiveness and feasibility of the proposed distribution method.

(4) However, the model does have certain limitations. First, its lack of a buffer against emergencies is an inevitable shortcoming: in practice, such a buffer should be provided. Second, determination of the distribution cycle was made based on short-term data; further comprehensive analysis based on long-term AFC data is expected in the future. Third, a quantitative expression of manpower can be included in the objective of the distribution model in the future research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (2020JBM046).

References

[1] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden Day, San Francisco, CA, USA, 2015.

[2] R. F. Engle and C. W. J. Granger, "Co-integration and error correction: representation, estimation, and testing, Econometrica," *Journal of the Econometric Society*, vol. 55, no. 2, pp. 251–276, 1987.

[3] Van Der, M. Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[4] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[5] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.

[6] M. C. Tan, S. C. Wong, and J. M. Xu, "An aggregation approach to short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 60–69, 2009.

[7] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6164–6173, 2009.

[8] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 3, pp. 211–234, 2005.

[9] Y. Wei and M.-C. Chen, "Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 148–162, 2012.

[10] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, pp. 568–581, 1964.

[11] H. Min, "The multiple vehicle routing problem with simultaneous delivery and pick-up points," *Transportation Research Part A: General*, vol. 23, no. 5, pp. 377–386, 1989.

[12] R. Bent and P. Van Hentenryck, "A two-stage hybrid local search for the vehicle routing problem with time windows," *Transportation Science*, vol. 38, no. 4, pp. 515–530, 2004.

[13] J. Wang, Y. Zhou, Y. Wang, J. Zhang, C. L. P. Chen, and Z. Zheng, "Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: formulation, instances, and algorithms," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 582–594, 2016.

[14] S. Mitra, "A parallel clustering technique for the vehicle routing problem with split deliveries and pickups," *Journal of the Operational Research Society*, vol. 59, no. 11, pp. 1532–1546, 2008.

[15] J. Dethloff, "Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up," *OR-Spektrum*, vol. 23, no. 1, pp. 79–96, 2001.

[16] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[17] G. Mosheiov, "Vehicle routing with pick-up and delivery: tour-partitioning heuristics," *Computers & Industrial Engineering*, vol. 34, no. 3, pp. 669–684, 1998.

[18] G. Jeon, H. R. Leep, and J. Y. Shim, "A vehicle routing problem solved by using a hybrid genetic algorithm," *Computers & Industrial Engineering*, vol. 53, no. 4, pp. 680–692, 2007.

[19] G. Martinovic, I. Aleksi, and A. Baumgartner, "Single-commodity vehicle routing problem with pickup and delivery service," *Mathematical Problems in Engineering*, vol. 2008, Article ID 697981, 17 pages, 2008.

[20] Y. Gajpal and P. Abad, "An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup," *Computers & Operations Research*, vol. 36, no. 12, pp. 3215–3223, 2009.

[21] J.-F. Chen and T.-H. Wu, "Vehicle routing problem with simultaneous deliveries and pickups," *Journal of the Operational Research Society*, vol. 57, no. 5, pp. 579–587, 2006.

[22] E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1070–1081, 2009.

[23] A. S. Tasan and M. Gen, "A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries," *Computers & Industrial Engineering*, vol. 62, no. 3, pp. 755–761, 2012.

[24] Beijing Subway co. LTD., *Management of ticket cards manual*, Beijing Subway co. LTD., Beijing, China, 2014, https://www.bjsubway.com/in Chinese.

[25] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Proceedings of the Second International Symposium on Information Theory*, Budapest: Akademiai Kiado, Tsahkadsor, Armenia, pp. 267–281, September 1973.

[26] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American Statistical Association*, vol. 74, no. 366a, pp. 427–431, 1979.

[27] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[28] A. Pankratz, *Forecasting with Univariate Box-Jenkins Models: Concepts and Cases*, Vol. 224, John Wiley & Sons, Hoboken, NJ, USA, 2009.

[29] Tiberlake, http://www.stata-uk.com/manuals.

[30] C. X. Liu, *Beijing Subway at Night and Off-Peak Hours for Urban Logistics System*, School of Economics, Beijing Supplies University, Beijing, China, 2011, in Chinese.

[31] W. S. Shen, "Reverie Beijing low carbon life and the development of subway freight function," *Beijing Observation*, vol. 1, pp. 13-14, 2010, in Chinese.

[32] A. Dampier and M. Marinov, "A study of the feasibility and potential implementation of metro-based freight transportation in Newcastle upon Tyne," *Urban Rail Transit*, vol. 1, no. 3, pp. 164–182, 2015.

[33] J. Kelly and M. Marinov, "Innovative interior designs for urban freight distribution using light rail systems," *Urban Rail Transit*, vol. 3, no. 4, pp. 238–254, 2017.

[34] Beijing Subway co. LTD., *Ticket Distribution Management Regulation*, Beijing Subway co. LTD., Beijing, China, 2014, in Chinesehttps://www.bjsubway.com/.

[35] L. U. Kai, M. H. Bao, L. U. Fang et al., "Urban rail transit in China: progress report and analysis (2008–2015)," *Urban Rail Transit*, vol. 2, no. 3-4, pp. 93–105, 2016.

[36] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[37] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 8403813, 10 pages
https://doi.org/10.1155/2020/8403813

Research Article

Performance of Polar-Coded 3D Image Transmission over Fading Channel

Jianjun Hao,[1] Luyao Liu,[1] and Wei Chen[2,3]

[1]*School of Electronic and Information Engineering, Shandong University of Science & Technology, Qingdao, Shandong 266590, China*
[2]*School of Computer Science and Technology, Mine Digitization Engineering Research Center of the Ministry of Education, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China*
[3]*School of Earth and Space Sciences, Peking University, Beijing 100871, China*

Correspondence should be addressed to Wei Chen; chenwdavior@163.com

Received 18 April 2020; Revised 20 May 2020; Accepted 27 May 2020; Published 16 June 2020

Guest Editor: Chi-Hua Chen

Any signal transmitted over an air-to-ground channel is corrupted by fading, noise, and interference. In this paper, a Polar-coded 3D point cloud image transmission system with fading channel is modeled, and also the simulation is performed to verify its performance in terms of 3D point cloud image data transmission over Rician channel with Gaussian white noise and overlap of Gaussian white noise + periodic pulse jamming separately. The comparison of Polar-coded scheme with RS-coded scheme in the same scenario indicates that Polar-coded system gives far better performance against AWGN noise and fading than the RS-coded system does in the case of short block length. But RS-coded scheme shows better performance on antipulse jamming than that of Polar-coded scheme, while there is no interleaving between codewords.

1. Introduction

Unmanned Aerial Vehicles (UAV) communication systems are more accessible to be applied in the civilian and military domains, such as building survey, target reconnaissance, surface mapping, precision agriculture, and postdisaster rescue. In these applications, it is often necessary to capture target images (videos) or 3D images [1, 2]. For example, in 3D detection of military targets, these 3D images need to be sent back to the ground station in real time, thus to guide the flight path or attitude adjusting by the drone operator. While transmitting on air-to-ground channel, these 3D images (videos) suffer from fading, noises, and interferences (such as radar signal jamming and a periodic strong pulse jamming) [3–5], and these will inevitably lead to bit errors at the receiver. A certain number of errors will cause changes in the shape and color of the 3D object and degrade the image quality.

Channel code can effectively reduce the transmission error rate and ensure reliable transmission against hostile channel conditions. Therefore, the application of channel code in a data transmission system can mitigate the effects of channel fading and noise interference, so as to improve the reliability of data transmission. In recent years, Polar code has been recognized and employed due to its good characteristics. In 2016, 3GPP was announced to adopt Polar code as the control channel code scheme for 5G mobile communication in eMBB scenario.

Polar code is a new channel coding method proposed by Erdal Arikan of Bilkent University, Turkey, in 2007. It is currently the only channel coding method which can theoretically reach the Shannon channel capacity. Compared with LPDC codes, Polar code scheme can achieve reliable communication with lower coding and decoding complexity.

Recent studies in Polar code employed in voice and image transmission achieved some valuable results. A voice transmission simulation adopting Polar code over AWGN and Rayleigh channel is performed in lecture [6], and the results show that this scheme provides high performance in

bit error correction. In [7], the transmission performance of Polar-coded and OFDM-modulated grayscale images is tested over AWGN channel; the results show that both PSNR and structural similarity (SSIM) are superior to the BCH code scheme in BER. In [8], the authors simulated Polar-coded image by passing through Rayleigh fading transmission system to test its performance on antifading. Zhao Shengmei and her team of Nanjing University of Posts and Telecommunications adopt combined Polar coding with OFDM scheme to test its performance on transmitting grayscale image over Rician fading channel. They drew a conclusion that, under the condition of the same code rate and code length, PSNR of received image can be improved by 4 dB compared to the LDPC coding scheme [9]. In [10], an unequal error protection scheme is proposed to alleviate the distortion of received image caused by transmission bit errors. The simulation results show that it provides 7 dB gain compared to the unsuitable UEP scheme. In lecture [11], a chaotic interleaving technique is proposed for improving a transmission of colored images over burst error environment through merging it with error control scheme, but this advantage is achieved at expense of complexity. In [12], the authors proposed a method to improve transmission speed of 3D point cloud by removing unnecessary information, but it achieved transmission rate improvement at the expense of reliability.

With the rise of VR (Virtual Reality), the processing and transmitting of 3D images and 3D video also attract many scholars' attention. Since data format and data structure of 3D images are quite different from 2D images, as well as the evaluation of image quality, transmitting 3D images becomes a big challenge. Aiming at solving the problem of transmitting images over mobile channels, we modeled a Polar-coded image transmission system with Rician fading channel to verify its performance on combating Gaussian noise and periodic strong pulse jamming and make a comparison with RS code scheme.

2. Polar Coding

The Polar-coding method is based on channel polarization which includes two processes: channel merging and channel splitting. N-independent binary symmetric discrete memoryless channels (BDMC) W can be regarded as a new joint channel W_N; we call this channel combining. After N original channels linearly combined and a joint channel generated, then the joint channel split into many new channels according to a certain rule; hence, a phenomenon called "channel polarization" occurs [11]. After the channel splitting, capacity of some split channels is equal to 0 (full-noise channel), and that of others is equal to 1 (noiseless channel), but the total channel capacity remains the same. According to this phenomenon, the data bits are then sent to transmitting over these noiseless channels [13, 14], so that the Shannon limit can be approached.

Polar codes can be encoded in the form of linear block codes. The mapping from the source code sequence u_1^N to the transition sequence x_1^N can be extracted and expressed by a matrix $x_1^N = u_1^N G_N$, and the original recursive equation of the generator matrix G_N is given by

$$G_N = (I_{N/2} \otimes F^{\otimes n}) R_N (I_2 \otimes G_{N/2}), \tag{1}$$

where $F^{\otimes n} = \begin{bmatrix} F^{\otimes n-1} & 0 \\ F^{\otimes n-1} & F^{\otimes n-1} \end{bmatrix}$ is generated by recursion of F, $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, I represents the identity matrix, and $\otimes$ is the Kronecker product; when $A_N = (I_{N/2} \otimes F) R_N$, equation (1) becomes $A_4 = A_N (I_2 \otimes G_{N/2})$. The recursive construction of the generator matrix can be implemented by generating A_N. For instance, given $N = 8$, we obtain

$$A_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

There are two parameters for the channels mentioned above, symmetric capacity $I(W)$ and the Bhattacharyya parameters $Z(W)$. $I(W)$ is the maximum information transmission rate of the channel, and $Z(W)$ is the minimum transmission error probability. One has

$$I(W) = \sum_{y \in Y} \sum_{x \in X} \frac{1}{2} W\left(\frac{y}{x}\right) lb \frac{W(y/x)}{(1/2)W(y/0) + (1/2)W(y/1)},$$
$$Z(W) = \sum_{y \in Y} \sqrt{W\left(\frac{y}{0}\right) W\left(\frac{y}{1}\right)}, \tag{3}$$

where $W(y/x)$ is the transition probability $X \longrightarrow Y$. After channel splitting, the capacity values of these channels are subjected to "0-1" distribution. Polarization coding is to select those channels with full channel capacity. A is defined as the selected channels' set for transmitting information data; K is the size of set A which represents the number of transmitted information data bits. Correspondingly, A^C is the set of channels employed to transmit the known bits; hence, the mapping code $x_1^N = u_1^N G_N$ can be decomposed into

$$x_1^N = u_A G_N(A) \oplus u_{A^C} G_N\left(A^C\right), \tag{4}$$

where $G_N(A)$ is equivalent to the information channel vector selected from G_N and $G_N(A^C)$ is the junk channel matrix composed of the remaining G_N rows, which determines the elements used to select the channel in the index set A. u_{A^C} is the default bit sequence value determined by the information source and destination [15]. Once the source message is given, the Polar code block x_1^N can be obtained by

constructing a set of (N, K, A, u_{A^C}). The process of polarization coding is as follows:

(1) Assuming that the coding length is $N = 2^n, n > 1$, calculate the generator matrix recursively according to formula $G_N = A_N (I_2 \otimes G_{N/2})$

(2) Set the coding efficiency value R and select the method of calculating the Bhattacharyya parameters under the BEC channel condition to obtain the full channel index set A

(3) Obtain generator matrix and channel index set; then select channels by these channel indexes to determine $G_N(A)$

(4) The sender and receiver agree in advance on the default bit sequence u_{A^C} and send it to the junk channel $G_N(A^c)$ for transmission

(5) After determining the set (N, K, A, u_{A^C}) in accordance with the previous four steps, calculate the coding sequence x_1^N to implement the polarization encoding

In this article, we use CRC-SCL decoding method to recover the image. The CRC-SCL decoding algorithm is based on the SCL decoding scheme by adding a cyclic redundancy check CRC block.

The process of CRC-SCL decoding algorithm is as follows: after the source generates K-bit information sequence, it is first sent to the CRC encoder to get m-bit check bit to attach to K bit information sequence; the obtained $k + m$ bits data block is then sent to the Polar encoder with a code length of N and a code rate of $(k - m)/N$; at the receiver, the received block first passes the SCL decoder, and the survived L decoding paths are verified by CRC check in order of large to small by metric norm. If the verification result is zero by CRC check, the path is considered to be correct and output. If no path passes the verification, the CRC-SCL algorithm is degraded to SCL algorithm.

3. Polar-Coded Image Transmission System Model

The image transmission system modeled in this paper adopts Polar code as the channel coding method and QPSK as modulation scheme. The bit rate is supposed to be 10 Mbps, carrier frequency 5 GHz, and total link power loss 20 dB. Considering that there is often a light sight propagation path between UAV and ground, we choose Rician fading channel with 3 propagation paths (including a direct path and 2 reflect paths with time delay 0.1 μs and 0.3 μs individually) as the wireless channel model. The noise is additive Gaussian white noise, and we use periodic strong jamming pulse (more than 5 dB higher than signal in mean power within jamming pulse duration) as pulse interference. The jamming pulse is presented as

$$J(t) = \eta_i \sum \left[g(t + nT) - g(t + nT + T_c) \right] e^{j2\pi f_0 (t+\tau_i)}, \quad (5)$$

where η_i is the amplitude of strong jamming pulse which follows Gaussian distribution and the random time delay $\mathrm{MSE}(a, b) = (1/(N \times M)) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} [a(x, y) - b(x, y)]^2$ follows uniform distribution. T is the pulse width, and $\mathrm{PSNR}(a, b) = 10\lg[255^2/\mathrm{MSE}(a, b)]$ denotes the period of jamming pulse. f_0 is the frequency of sine signal in jamming pulse duration, and $g(t)$ represents rectangular pulse. The image transmission system model is illustrated in Figure 1.

We established a 3D image transmission system model with Rician fading channel by MATLAB m-file. Polar code scheme with code length of 1024 and coding efficiency of 1/2 is employed to encode the message under the condition of BEC channel with a deletion probability $p = 0.5$ and 2 decoding paths. After being modulated by $\pi/4$-QPSK, 2D image and 3D image signals transmit over a Rician fading channel with Gaussian noise, periodic strong pulse jamming, and Gaussian noise + periodic strong pulse jamming, respectively, to verify the performances of 3D image transmission. Also a RS-coded scheme (BM decoding) with coding efficiency 0.5 is performed over a Rician channel to compare with Polar-coded system in transmission performances.

In this article, the demonstration scheme is passing the 2D image data and the 3D point cloud image data through the established transmission system. The transmitted images are examined for their visual quality by using two different image quality assessment metrics, peak signal-to-noise ratio (PSNR, for 2D image) and point error ratio (PER, for 2D point cloud).

4. Simulation of 2D Image Passing through the Polar-QPSK System

In 2D image transmission simulation, a grayscale image Lena is subjected to Polar/RS encoding and then modulated by QPSK modulator; the output signals are sent over a Rician fading channel with Gaussian white noise to verify the performance against fading and Gaussian white noise.

Figures 2(a)–2(c) are the Polar-coded images reconstructed at receiver after passing through Polar/RS-coded transmission system with Rician fading channel ($k = 4$) and Gaussian noise. The received image and its PSNR under different SNR (average signal-to-noise ratio at the receiver) are shown in Figures 3(a)–3(c). The equation for calculating PSNR is

$$\mathrm{MSE}(a, b) = \frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} [a(x, y) - b(x, y)]^2,$$
$$\mathrm{PSNR}(a, b) = 10 \lg \left[\frac{255^2}{\mathrm{MSE}(a, b)} \right]. \quad (6)$$

In order to study the influence on image quality caused by pulse jamming, we use a nonfading channel to carry out the simulation. Figures 4(a) and 4(b) are the reconstructed Polar-coded images at receiver after passing through a channel without fading but interfered by periodic strong jamming pulse (with period of 1.64 ms, i.e., 8 codewords' width). Figures 4(c) and 4(d) show the reconstructed RS-coded images under same conditions. Here, JSR is the ratio of the average power of the jamming pulse to the average

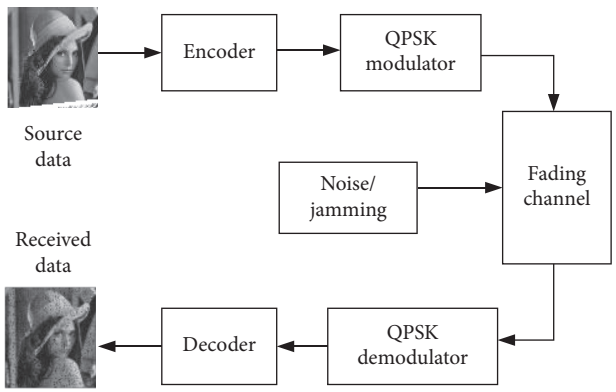


Figure 1: Polar-coded image transmission system model.

power of the signal in pulse duration, and the jamming pulse width is 12.8 μs (64 symbols). For a strong jamming pulse (JSR > 4 dB), jamming pulse amplitude's variation has a little effect on Polar-coded image or RS-coded image quality, as shown in Figure 4. Figures 4(a)–4(d) also indicate that RS code scheme provides a better performance on bursts error caused by strong pulse jamming than Polar code does.

Figures 5(a)–5(c) are the reconstructed Polar-coded images passed through the fading channel (Rician channel, $k = 2$) with Gaussian white noise and periodic pulse jamming (jamming pulse width of 10.24 μs). It can be seen that while SNR increases, the image PSNR improves rapidly, but when SNR is up to a certain value (1 dB in this experiment), there is no longer evident evaluation in PSNR. Although the RS code has good antipulse jamming performance, it cannot achieve a high image quality at low SNR (Figures 6(a)–6(c)).

5. Simulation of 3D Point Cloud Passing through the Polar-QPSK System

In 3D image transmission performance verification experiment, we extract point cloud data from a 3D image file--teapot.obj, and then these image data are sent to pass through Polar-coded and RS-coded 3D image system, respectively.

Figure 7(a) is the original image in obj format, and Figure 7(b) is the extracted 3D point cloud without color parameter.

In this paper, we use the point filling method to convert the extracted point cloud data to bit streams for transmission.

3D point cloud image data are processed in steps as follows:

(1) Quantize coordinate value of the point to an integer; the x coordinate range is from to 1 to 141, y from 1 to 70, and z from 1 to 86.

(2) According to the value of z, the 3D point cloud is divided into 86 layers (planes). In each layer, those positions on each plane with a coordinate value are set to 1, and the others are set to 0; thus, a $141 * 70 * 86$ 3D matrix with elements "0" or "1" is generated.

(3) Reconstruct this 3D matrix into a one-dimensional data sequence, thus generate a bit stream of 848820 bits, and hence fill the bit stream to a length of $2n$ and send it to the encoder of the transmission system.

(4) At receiver after decoding the bit stream at the receiving end, the decoded sequence is then reconstructed into a 3-dimensional matrix of $141 * 70 * 86$. In each layer, when a bit is decided as 1, it means there is a voxel at the corresponding position, and the value is taken out to reconstruct the 3D point cloud image.

Figures 8(a), 8(b), 9(a), and 9(b) are the reconstructed 3D point clouds after passing through Polar-coded and RS-coded system, respectively. The channel in transmission system adopted Rician fading channel ($k = 2$) with Gaussian white noise. Figure 10 shows the plot of point error ratio (PER) versus SNR; it can be seen that the Polar-coded cloud image quality improves rapidly with the increase of signal-to-noise ratio, whereas point error ratio curve of RS scheme depicts that there is no evident improvement in point cloud quality with SNR increase. Here, PER is the ratio of the amount of error points to total $141 * 70 * 86$ points. It indicates that the performance against fading and Gaussian white noise of Polar-coded system is evidently better than that of RS-coded system.

Figures 11(a)–11(d) and 12(a)–12(d) are the recovered 3D point clouds at the receiving end of Polar-coded system and RS-coded system, respectively.

Point clouds are also sent to pass through Polar/RS system with Rician fading channel ($k = 2$), Gaussian white noise, and periodic pulse jamming (12.8 μs of pulse width, 5 dB of JSR). The amount of error points of total $141 * 70 * 86$ points denoted by M in figures is the evaluation index of recovered image quality. Figures 11(a)–11(d) and Figure 13 illustrate that the quality of point cloud evaluates rapidly with the increase of the signal-to-noise ratio, but while SNR exceeds 2 dB, the error points change very little with SNR. When SNR = 4, the error points number M is only 48 less than that of SNR = 2, and the amount of error points at SNR = 6 is only 308 less than that of SNR = 4.

When SNR < 4, Polar-coded transmission system shows far better performance than that of RS coding system on combating fading, AWGN noise, and pulse jamming, but when SNR is up to 6, the error points of recovered teapot point cloud of RS-coded system decline to 30, which means that its quality is very close to the original picture, as shown in Figure 13.

Figures 14(a), 14(b), 15(a), and 15(b) are the received 3D point clouds of Polar-coded scheme and RS-coded scheme via nonfading channels with periodic strong pulse interference, where Figures 14(a) and 15(a) are the received images interfered by a strong pulse with width of 12.8 μs (64 symbols), and Figures 14(b) and 15(b) are the received images interfered by a strong pulse with width of 51.2 μs. As

(a) (b) (c)

Figure 2: Recovered Polar-coded image (transmitted over Rician fading channel with Gaussian noise). (a) SNR = −2 dB; PSNR = 30.60. (b) SNR = −1 dB; PSNR = 37.09. (c) SNR = 0 dB; PSNR = 46.63.

(a) (b) (c)

Figure 3: Recovered RS-coded image (transmitted over Rician fading channel-added Gaussian noise). (a) SNR = −2 dB; PSNR = 32.87. (b) SNR = −1 dB; PSNR = 34.62. (c) SNR = 0 dB; PSNR = 36.72.

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

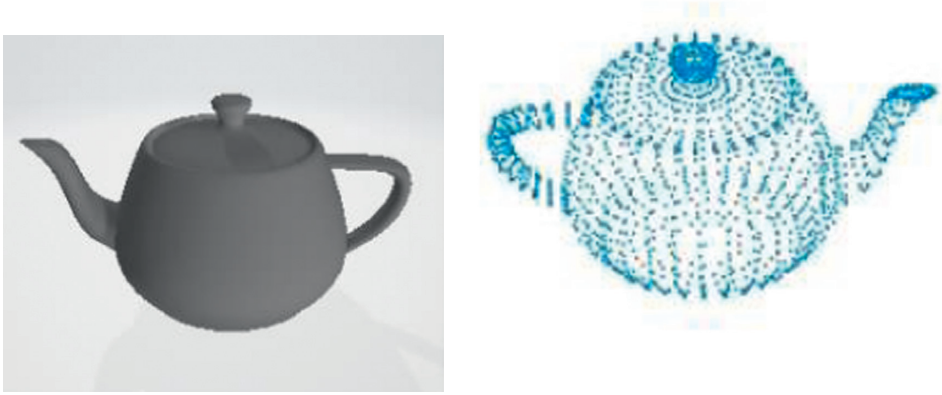
Figure 4: Recovered Polar-coded image and RS-coded image (with periodic strong pulse jamming). (a) JSR = 5 dB; PSNR = 38.5. (b) JSR = 10 dB; PSNR = 38.4. (c) JSR = 5 dB; PSNR = inf. (d) JSR = 10 dB; PSNR = inf.

(a) (b) (c)

Figure 5: Recovered Polar-coded images (transmitted over fading channel-added periodic strong pulse jamming). (a) SNR = −2; Psnr = 30.6215. (b) SNR = 0; psnr = 38.1989. (c) SNR = 2; psnr = 39.5724.

(a) (b) (c)

Figure 6: Received RS-coded images (transmitted over fading channel-added periodic strong pulse jamming). (a) SNR = −2; PSNR = 26.48. (b) SNR = 0; PSNR = 28.79. (c) SNR = 2; PSNR = 30.4.

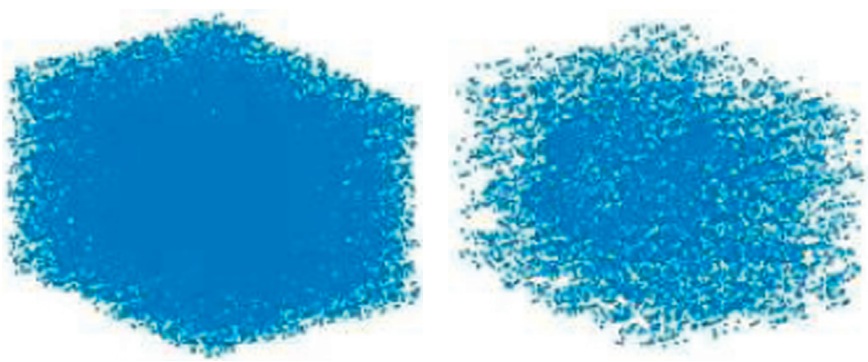


(a) (b)

Figure 7: 3D image of the teapot. (a) Original obj image. (b) Extracted 3D point cloud.

(a) (b)

Figure 8: Reconstructed 3D point cloud (Polar-coded system with fading channel and AWGN noise). (a) SNR = 0 dB; (b) SNR = 2 dB.



(a) (b)

Figure 9: Reconstructed 3D point cloud (RS-coded system with fading channel and AWGN noise). (a) SNR = 0 dB; (b) SNR = 2 dB.

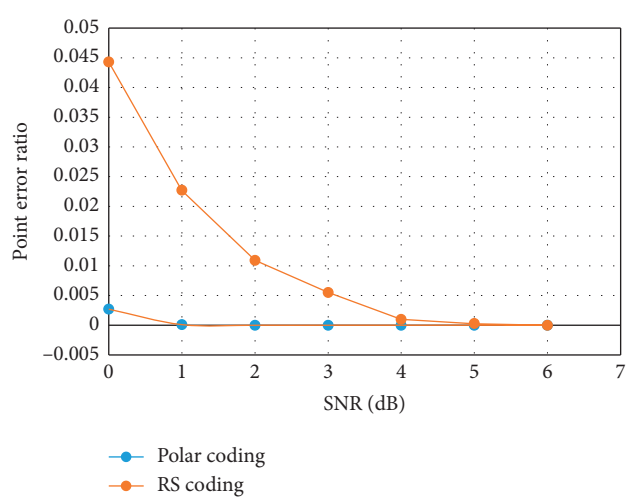


Figure 10: Point error ratio versus SNR (Polar/RS system with fading channel and AWGN noise).

can be seen from Figure 14, a wide pulse width (pw) leads to a worse image quality (a large M value).

It can be seen from Table 1 that when jamming pulse width is within a certain limit (less than 51.2 μs), strong jamming pulses have no effect on the quality of the received RS-coded point cloud image. Compared to Polar, it indicates that the performance of RS code scheme is better than that of Polar code scheme on antistrong pulse jamming.

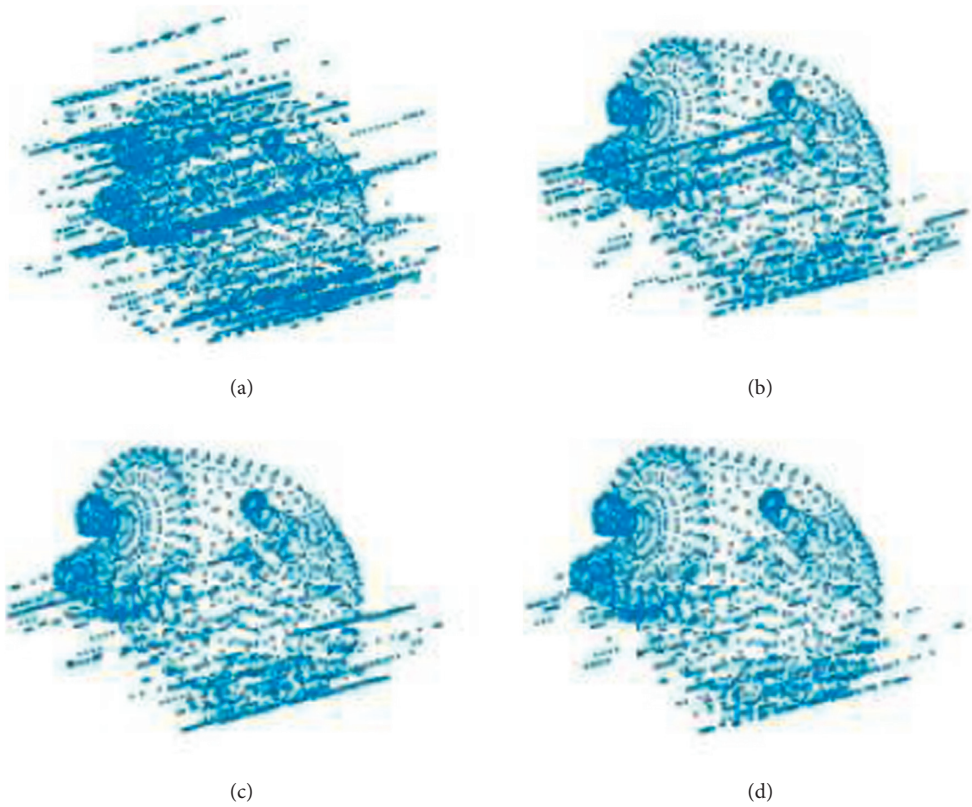


Figure 11: Reconstructed 3D point cloud (Polar-coded system with fading channel and AWGN noise and periodic strong pulse jamming). (a) SNR = 0 ($M = 4069$); (b) SNR = 2 ($M = 1100$); (c) SNR = 4 ($M = 1052$); (d) SNR = 6 ($M = 744$).

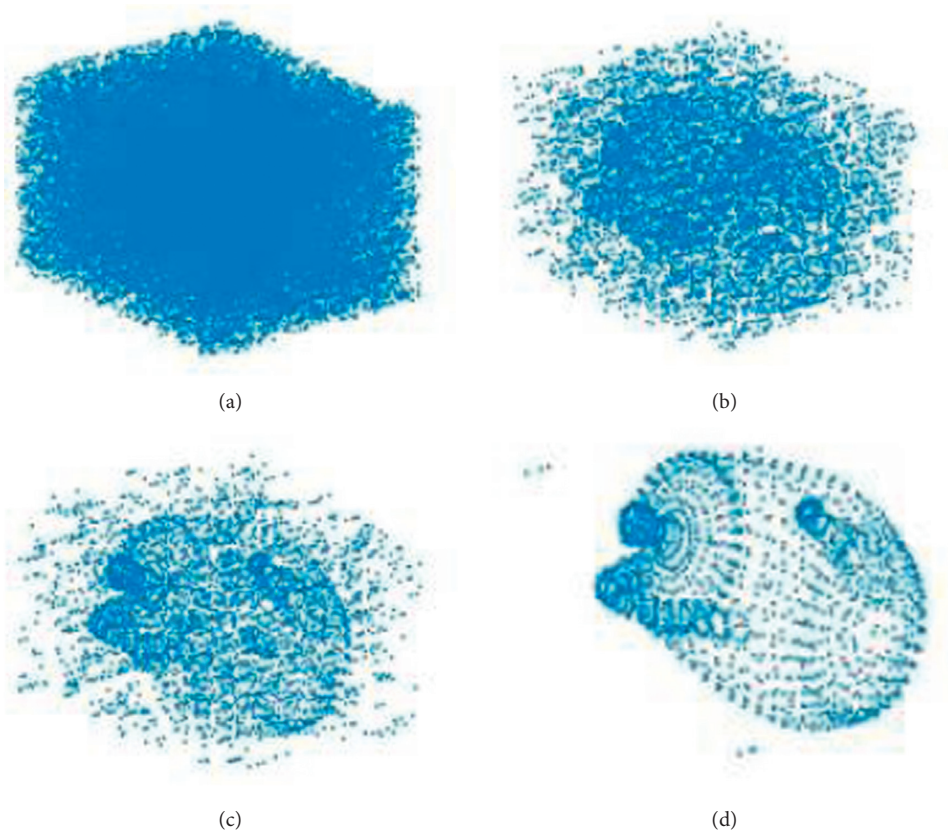


Figure 12: Reconstructed 3D point cloud (RS-coded system with fading channel and AWGN noise and periodic strong pulse jamming). (a) SNR = 0 ($M = 37387$); (b) SNR = 2 ($M = 8820$); (c) SNR = 4 ($M = 1486$); (d) SNR = 6 ($M = 30$).

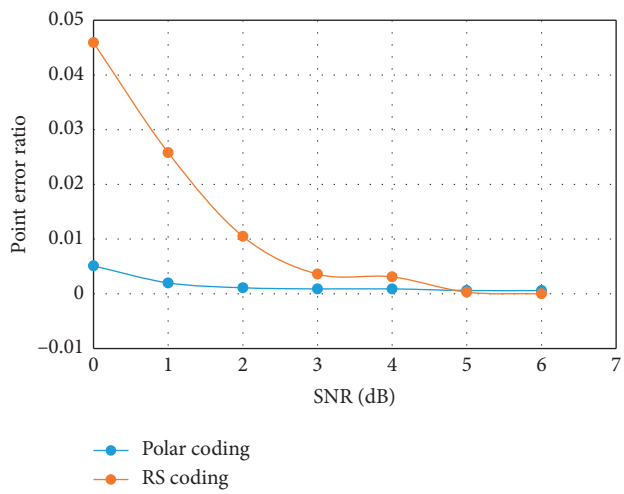
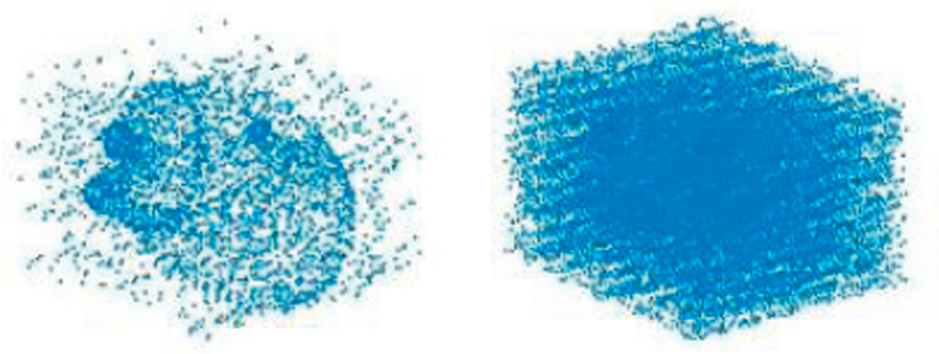


Figure 13: Point error ratio versus SNR (Polar/RS system with fading channel, AWGN noise, and strong pulse jamming).



(a) (b)

Figure 14: Recovered Polar-coded 3D point cloud (with periodic strong pulse jamming). (a) pw = 12.8 μs ($M = 1572$); (b) pw = 51.2 μs ($M = 18336$).

(a) (b)

Figure 15: Recovered RS-coded 3D point cloud (with periodic strong pulse jamming). (a) Pw = 12.8 μs ($M = 0$); (b) pw = 51.2 μs ($M = 0$).

Table 1: Multimodel comparison.

Pulse width (μs)	Error points (M) (polar code)	Error points (M) (RS code)
6.4 (32 symbol)	373	0
12.8 (64 symbol)	1572	0
25.6 (128 symbol)	6299	0
51.2 (256 symbol)	18336	0
76.8 (384 symbol)	28216	106745
102.4 (512 symbol)	42714	140733

6. Conclusion

We established an image transmission system model based on Polar/RS code scheme and then tested its performances on combating channel fading, additive Gaussian white noise, and periodic pulse jamming by transmitting 2D image and 3D point cloud image, respectively. Simulation results indicate that Polar-coded system provides better performances on antifading and anti-AWGN noise than RS system does, whereas RS system performs well on antiperiodic pulse jamming. Simulation results also show that the improvement of JSR of strong jamming pulse has a little impact on received image quality, while the increase in width of strong jamming pulse degrades the 3D point cloud quality evidently.

Data Availability

The .xls data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by the National Natural Science Foundation of China, under Grant nos. 51874300 and 61801270, Shanxi Provincial People's Government Jointly Funded Project of China for Coal Base and Low Carbon, under Grant no. U1510115, and the Open Research Fund of Key Laboratory of Wireless Sensor Network and Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, under Grant nos. 20190902 and 20190913.

References

[1] S. Siebert and J. Teizer, "Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system," *Automation in Construction*, vol. 41, pp. 1–14, 2014.

[2] L. Comba, A. Biglia, D. Ricauda Aimonino, and P. Gay, "Unsupervised detection of vineyards by 3D point-cloud UAV photogrammetry for precision agriculture," *Computers and Electronics in Agriculture*, vol. 155, pp. 84–95, 2018.

[3] D. W. Matolak, "3D Air-X UAV communications: challenges and channel modeling," in *Proceedings of the International Workshop on Communication Technologies for Vehicles,* Lecture Notes in Computer Science, Moreno, CA, USA, May 2018.

[4] H. S. Silvaa, M. S. Alencarb, W. J. L. Queirozc, D. B. T. Almeida, and F. Madeiro, "Bit error probability of M-QAM under η-μ or κ-μ fading and impulsive noise gated by a signal characterized by a Markov process or Poisson process," *Digital Signal Processing*, vol. 100, pp. 1–14, 2020.

[5] Y. Li, W. Wang, H. Gao et al., "Air-to-ground 3D channel modeling for UAV based on Gauss-Markov mobile model," *AEU—International Journal of Electronics and Communications*, vol. 114, Article ID 152995, 2020.

[6] S. Zhao, P. Shi, and B. Wang, "Polar codes and its application in speech communication," in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–4, IEEE, Nanjing, China, November 2011.

[7] A. Mishra, K. Sharma, and A. De, "Quality image transmission through AWGN channel using polar codes," *International Journal of Computer Science and Telecommunications*, vol. 5, no. 1, 2014.

[8] P. Shi, Z. Zhifang, and L. Gong, "Studies on performance of polar coded image transmission," *Journal of Nanjing University of Post & Telecommunication (Natural Science)*, vol. 34, no. 2, pp. 65–71, 2014.

[9] H. Fan, S. Zhao, and B. Zheng, "Construction and analyses corresponding performance of polar codesin rician channels," *Journal of Signal Processing*, vol. 31, no. 9, pp. 1062–1066, 2015.

[10] M. A. M. Mohamed El-Bendary, A. E. Abou El-Azm, N. A. El-Fishawy et al., "JPEG image transmission over mobile network with an efficient channel coding and interleaving," *International Journal of Electronics*, vol. 99, no. 11, Article ID 1491518, 2012.

[11] C. Yang, Z. Wang, W. He, and Z. Li, "Development of a fast transmission method for 3D point cloud," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 25369–25387, 2018.

[12] D. Wu, Y. Li, and Y. Sun, "Construction and block error rate analysis of polar codes over AWGN channel based on Gaussian approximation," *IEEE Communications Letters*, vol. 18, no. 7, pp. 1099–1102, 2014.

[13] S. Kahraman and M. E. Celebi, "Code based efficient maximum-likelihood decoding of short polar codes," in *Proceedings of the 2012 IEEE International Symposium on Information Theory Proceedings*, pp. 1967–1971, IEEE, Cambridge. Massachusetts, USA, July 2012.

[14] I. Tal and A. Vardv, "List decoding of polar codes," in *Proceedings of the 2011 IEEE International Symposium on Information Theory Proceedings*, IEEE, St. Petersburg, Russia, July 2011.

[15] K. Chen, K. Niu, and J. R. Lin, "Improved successive cancellation decoding of polar codes," *Electronics Letters*, vol. 61, no. 8, pp. 3100–3107, 2012.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 5761414, 15 pages
https://doi.org/10.1155/2020/5761414

Research Article

Fast Vehicle and Pedestrian Detection Using Improved Mask R-CNN

Chenchen Xu,[1] Guili Wang,[1,2] Songsong Yan,[1] Jianghua Yu,[3] Baojun Zhang,[1] Shu Dai,[1] Yu Li,[1] and Lin Xu[4]

[1]*School of Physics and Electronic Information, Anhui Normal University, Wuhu 241002, China*
[2]*Anhui Provincial Engineering Laboratory on Information Fusion and Control of Intelligent Robot, Wuhu Anhui, 241002, China*
[3]*School of Communications and Information Engineering, Xi'an University of Posts & Telecommunications, Xi'an 710061, China*
[4]*School of Mathematics and Statistics, Anhui Normal University, Wuhu, 241002, China*

Correspondence should be addressed to Guili Wang; xlpyhwgl@ahnu.edu.cn

Received 16 February 2020; Revised 16 April 2020; Accepted 24 April 2020; Published 31 May 2020

Guest Editor: Feng-Jang Hwang

This study presents a simple and effective Mask R-CNN algorithm for more rapid detection of vehicles and pedestrians. The method is of practical value for anticollision warning systems in intelligent driving. Deep neural networks with more layers have greater capacity but also have to perform more complicated calculations. To overcome this disadvantage, this study adopts a Resnet-86 network as a backbone that differs from the backbone structure of Resnet-101 in the Mask R-CNN algorithm within practical conditions. The results show that the Resnet-86 network can reduce the operation time and greatly improve accuracy. The detected vehicles and pedestrians are also screened out based on the Microsoft COCO dataset. The new dataset is formed by screening and supplementing COCO dataset, which makes the training of the algorithm more efficient. Perhaps, the most important part of our research is that we propose a new algorithm, Side Fusion FPN. The parameters in the algorithm have not increased, the amount of calculation has increased by less than 0.000001, and the mean average precision (mAP) has increased by 2.00 points. The results show that, compared with the algorithm of Mask R-CNN, our algorithm decreased the weight memory size by 9.43%, improved the training speed by 26.98%, improved the testing speed by 7.94%, decreased the value of loss by 0.26, and increased the value of mAP by 17.53 points.

1. Introduction

To improve driving safety and reduce driver fatigue, research is being conducted on the development of intelligent driving technology [1]. In intelligent driving, we need to first guarantee the human's safety, and therefore, the assisted driving system (ADS) [2] to improve safety is a hot spot in intelligent driving research. The collision avoidance warning system (CAWS) [3] is particularly important for ADS in smart cars. One key issue of CAWS is the awareness of the driver's surroundings. The images of vehicles and pedestrians captured by car cameras are to be identified, detected, and divided by object detection technology, which faces challenges due to complex scene information.

The two main methods for vehicles and pedestrians detection are machine learning-based [4] approaches and deep-learning-based [5] approaches. Machine learning approaches first define features using one of the feature acquisition descriptors such as histogram of oriented gradient (HOG) [6] and then perform classification using a technique such as a support vector machine (SVM) [7]. The HOG + SVM approach shows superior performance but suffers from low mean average precision (mAP) and is not suitable for multistage process feature extraction [8]. Deep learning systems, such as convolutional neural networks (CNNs), show superiority in object detection because they aim to discover discriminative features from raw data [9]. The CNN was developed in the 1980s and 1990s [10], but since experiencing a resurgence of interest [11] in 2012, it has established a foothold in the field of computer vision and has grown at a rapid pace.

As the requirements for algorithm accuracy and speed continue to increase, vehicle and pedestrian recognition

algorithms such as R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN have been proposed [12–15]. Mask R-CNN was proposed in 2017. Without adding any skills, Mask R-CNN outperformed all the single recognition models at the time and defeated the 2016 champion in the Microsoft COCO dataset [16] challenge. It became the leader in image recognition, detection, and segmentation [15]. In view of the fact that Mask R-CNN has the advantages of fast speed and high accuracy in target detection tasks, it is applied to many fields [17, 18].

Mask R-CNN consists of three parts: feature pyramid network (FPN), regional proposal network (RPN), and detection. It can accomplish three tasks: target recognition [19], detection [20], and segmentation [21]. Mask R-CNN is dramatically driving the development of computer vision, leading to a series of proposed algorithms. The FPN is the first part of Mask R-CNN, which uses Resnet-101 [22] as its backbone to detect 81 types of targets. The depth of network structure is very important for many visual recognition tasks. Resnet is a residual learning framework to ease the training of networks that reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions.

Aiming at the research target of this paper, which is vehicles and pedestrians on the road, the following four improvements are made based on Mask R-CNN. First, in order to obtain more precise feature semantic information combined with the characteristics of semantic feature graph information, the side fusion calculation is added in the FPN structure, and the Side Fusion FPN (SF-FPN) algorithm is proposed. The second is to improve the Resnet-101 network to Resnet-86 network. The residual block is reduced from 23 to 18, which effectively improves the calculation speed. The third is to select 500 RPN training frames and 250 prediction frames to further reduce algorithm redundancy. The fourth is to classify and supplement the 81 samples in the COCO dataset [23] and change the rectangular label in the original image to a polygon label to improve the training effect.

2. Mask R-CNN

Mask R-CNN is a conceptually simple, flexible, and general framework for object recognition, detection, and instance segmentation, which can efficiently detect objects in an image, while generating a high-quality segmentation mask for each instance. Feature pyramid networks (FPNs) for object detection [24], the first block structure of Mask R-CNN, are responsible for feature extraction. The regional proposal network (RPN) [25], the second piece of Mask R-CNN, shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals [26]. We then expanded the Faster R-CNN to form the Mask R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition.

The RPN was applied to Mask R-CNN instead of selective search [27] so that the RPN can share the convolution feature of the full map with the detection network. It can predict both boundary position and object scores at each location, and it is also a fully convolutional network (FCN) [28]. As shown in Table 1, the Faster R-CNN uses the RPN as a region generation network to generate candidate regions. The FPS based on the Fast R-CNN algorithm is as high as 5, and its MAP tested on VOC 2012 is also increased to 70.4% [14].

To further improve the detection accuracy of the target, Mask R-CNN uses the bilinear interpolation algorithm region of interest (ROI) align instead of ROI pool [29] on the basis of Faster R-CNN. The ROI align layer removes the harsh quantization of the ROI pool and properly aligns the extracted features with the input. This method of ROI align avoids any quantization of the ROI boundaries or bins. The algorithm of ROI align is used to compute the exact values of the input features based on bilinear interpolation [30] at four regularly sampled locations in each ROI bin and aggregate the results. This method improves the accuracy of Mask R-CNN by 10% [15].

In order to enable Mask R-CNN to implement the mask function, Mask R-CNN adds mask branches to achieve high-precision instance segmentation from pixel-to-pixel alignment. Mask R-CNN can accomplish three tasks: target recognition, detection, and segmentation. Its detection speed can still reach 5 FPS. The flowchart of Mask R-CNN is shown in Figure 1. At the input, after the image passes through the FPN, five sets of feature maps of different sizes are generated, and the candidate frame area is generated by the RPN. After the candidate region is combined with the feature map, the system can achieve the detection, classification, and mask of the target. To further improve the computing speed of the algorithm, it can adapt to the real-time requirements of the intelligent driving anticollision warning system.

Based on Mask R-CNN, we propose a method to improve the detection of accuracy and speed through SF-FPN with Resnet-86. In this study, the dataset, FPN structure, and RPN parameter settings are improved. The improved method proposed in this study can realize the recognition, detection, and segmentation of the target at the same time.

3. Improvement Based on Mask R-CNN

3.1. Feature Pyramid Networks for Object Detection (FPN). Feature extraction is an important part of the field of machine vision. With the development of machine learning, methods based on neural network feature extraction, including featurized image pyramid [31], single feature map [32], pyramidal feature hierarchy [33], and feature pyramid network (FPN), have been proposed.

As shown in Figure 2(a), featurized image pyramid refers to an image input that passes through different convolutional layers, sets convolution kernels of different sizes, generates multiscale feature maps, and outputs feature maps of different sizes. Although this method obtains feature maps at different scales, it adds a large amount of calculation time and the semantic information from the feature map is not sufficient. Figure 2(b) shows a method of single feature map feature extraction. The idea is to input an image and pass different convolutional layers from bottom to top, with the

Table 1: Frame comparison of R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN.

Network framework	R-CNN	Fast R-CNN	Faster R-CNN	Mask R-CNN
Propose time	2014	2015	2016	2017
Region proposal	Selective search	Selective search	RPN	RPN
Feature extraction	CNN	CNN + ROI pool	CNN + ROI pool	CNN + ROI align
Feature classification	SVM			
Function	Classification, detection	Classification, detection	Classification, detection	Classification, detection, segmentation
Test time per image (with proposals)	47 seconds	2 seconds	0.2 seconds	0.2 seconds
mAP (VOC 2012)	62.4%	68.4%	70.4%	—

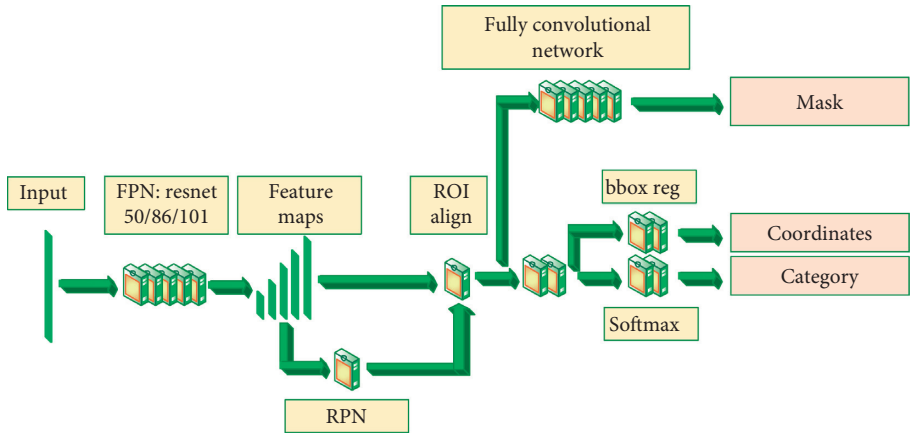


Figure 1: The Mask R-CNN framework flowchart.

output of the last convolutional layer used as the final feature output of the network. This method offers faster operation and utilizes the semantic information on each layer. For this reason, it has been previously applied to SPP-net, Fast R-CNN, and Faster R-CNN. But the performance of multiscale target detection is poor. Figure 2(c) shows that the pyramidal feature hierarchy method is still about inputting a picture from the bottom to top and passing through different convolutional layers. But it extracts different scale features of different layers as predictions, which will not increase the amount of calculation and can obtain multiscale features. Although the pyramidal feature hierarchy method can maintain speed and generate multiscale feature information at the same time, this method can neither make full use of lower level semantic information nor achieve good results of small target detection. To this end, the FPN adds a side link algorithm based on the pyramidal feature hierarchy algorithm, that is, when inputting one image, it passes through different convolutional layers from the bottom up and then links from top-down at side and combines low resolution and strong semantic features with high resolution and weak semantic features. It not only maintains the original calculation speed but also generates accurate multiscale feature information.

The top-down pyramid [34] algorithm in Figure 2(d) uses convolutional sampling to first reduce the size and then uses upsampling to increase the feature map. The network has no horizontal connection, that is, the top-down process does not integrate the original features, which will cause the location characteristics of the target to become more inaccurate after multiple downsampling and upsampling processes. The finest level only [35] algorithm shown in Figure 2(e) is capable of taking only the last layer P2 of PFN as the output and does not produce multiscale output. The sliding of the RPN stage window at different layers of the pyramid will increase the robustness of the scale change, so the FPN is useful for identifying different sizes of the target in robustness, which is significantly better than the finest level only algorithm.

As can be seen from Table 2, compared to the outputs of C4 and C5 of the featured image pyramid algorithm, the FPN algorithm improves the accuracy by nearly 21.7%. Particularly in the small target detection, it increases a significant advantage of 12.9 points. From Figure 2, we can also see that the FPN adds top-to-bottom side links and multiscale output compared with the single feature map and pyramidal feature hierarchy algorithm. In this way, low resolution and strong semantic features can be fully integrated with high resolution and weak semantic features. From the data table, we can see that when 1000 anchors are generated as predictions, the AR is improved by 6.8 points on average, of which 14.4 points are improved on the detection result of small targets. Compared with the top-down pyramid algorithm, the FPN adds horizontal connection and multiscale output, which increases 10.2 points in AR, including 18.4 points for small target detection. Finally, compared with the final level algorithm, FPN increases multiscale output and improves 4.0 points in AR. The small

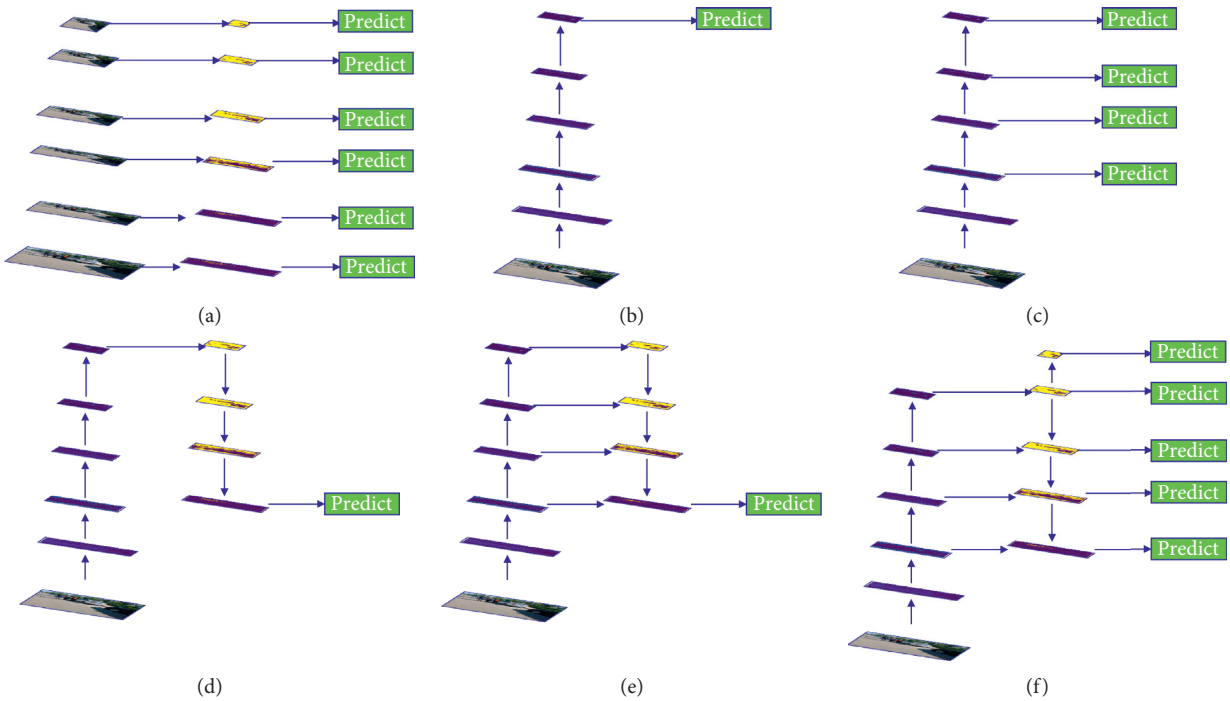


Figure 2: Six network structures for feature extraction. (a) Featurized image pyramid. (b) Single feature map. (c) Pyramidal feature hierarchy. (d) Top-down pyramid. (e) Finest level only. (f) Feature pyramid network.

Table 2: Feature extraction network data comparison.

Network structure	Feature	Anchors	Lateral	Top-down	AR^{1k}	AR_s^{1k}	AR_m^{1k}	AR_l^{1k}
Featurized image pyramid	C4	47k	No	No	48.3	32.0	58.7	62.2
Featurized image pyramid	C5	12k	No	No	44.9	25.3	55.5	64.2
Bottom-up pyramid	$\{P_k\}$	200k	Yes	No	49.5	30.5	59.9	68.0
Top-down pyramid	$\{P_k\}$	200k	No	Yes	46.1	26.5	57.4	64.7
Finest level only	P2	200k	Yes	Yes	51.3	35.1	59.7	67.6
FPN	$\{P_k\}$	**200k**	**Yes**	**Yes**	**56.3**	**44.9**	**63.4**	**66.2**

The first column is the feature extraction algorithm, where the second and third rows are the two-layer outputs of the algorithm, and the fourth row contains the single feature map and pyramidal feature hierarchy. The second column is the name of the output layer, where the "{}" symbol indicates the independent prediction at each layer. The evaluation standard uses average recall (AR). The number in the upper right corner of the AR indicates the number of anchors generated by each image. The letters "s," "m," and "l" in the bottom right corner denote the small goal, medium goal, and large goal, respectively.

targets have improved by 9.8 points, which greatly improves the robustness.

The FPN in Mask R-CNN uses Resnet-101 as the backbone. The design of the deep residual network overcomes the problem that the learning efficiency becomes lower due to the deepening of the network and the inability to effectively improve accuracy. The deep residual network divided a training series into one block for training so that the error of each block is minimized in order to achieve the goal of the smallest overall error. Resnet-101 is an internationally used classical deep residual network. It can be roughly divided into five stages of convolutional layers. The output scale is reduced by half at each stage. This FPN + Resnet network has robustness and adaptability and can not only send high-level features to low-level features but also make full use of all high-level feature information and underlying feature information through side links, thereby improving feature extraction capabilities. The FPN is the first part of Mask R-CNN, which can obtain feature maps. As shown in Figure 3, the FPN is built on the basis of image pyramids [36]. Its input image is obtained from the convolutional layer to obtain five sets of characteristic maps (C1, C2, C3, C4, and C5), and all five are upconverted or reduced to 256 dimensions by a $1 * 1$ size convolution kernel. Since the upsampling of C5 is as same as the one of C4, we thus use the dimensionality reduction results of C4 directly. This connection method can connect the high-level features of low resolution and high-semantic information and the low-level features of high resolution and low-semantic information from top to bottom so that the features of all scales have rich semantic information. The same is true for the connection method of P5, P4, P3, and P2. In order to make the large target detection effect better, P5

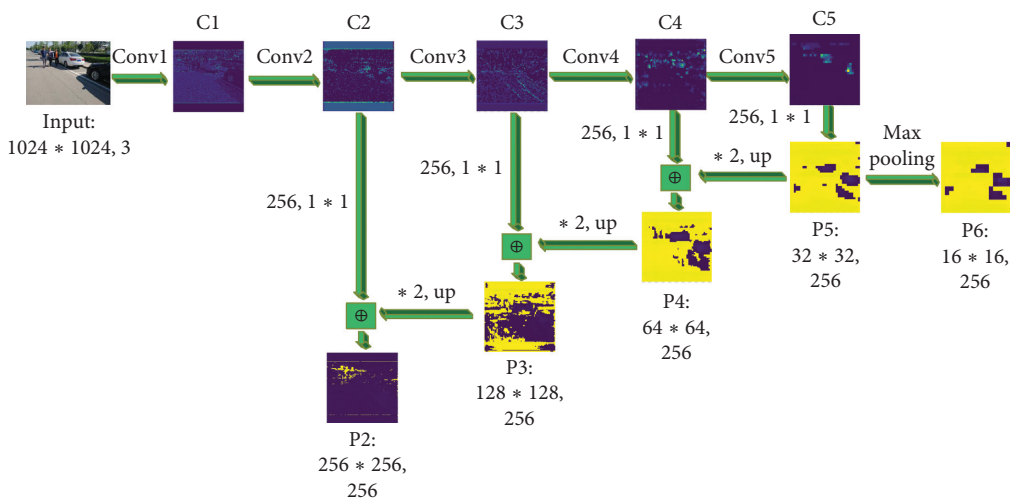


Figure 3: Network structure diagram of FPN + Resnet.

performs the maximum pooling at the end and forms a feature map P6 of 16 ∗ 16 size.

The COCO datasets contain 81 categories. The FPN in Mask R-CNN uses Resnet-101 as the backbone to detect 81 types of targets. However, since the detection target in this study consists of only three categories—person, car, and bus—there is a problem of parameter redundancy when using Resnet-101 to detect three types of targets. To reduce the redundancy and improve the computing speed, this study designs a Resnet-86 with only 86 layers as the backbone of the network and FPN to detect the three types of targets. As can be seen from Table 3, the Resnet-86, Resnet-50, and Resnet-101 structures are all composed of five-part convolutional layers. The number of residual blocks of Resnet-50 and Resnet-101 at Conv_4 are 6 and 23, respectively. In this study, the number of residual blocks of Resnet-86 at Conv_4 is changed to 18. It can be seen from the experimental results in Table 4 and Figure 4 that although the Resnet-50 structure is faster in recognition speed, its recognition accuracy cannot meet our requirements. Compared with Resnet-101, Resnet-86 not only increases the computing speed by about 7.94% but also reduces the weight memory by 9.43%. This can effectively promote the development of deep learning in the field of embedded development. Therefore, this study uses Resnet-86 as the backbone of Mask R-CNN to extract the features of the picture.

3.2. Side Fusion FPN (SF-FPN). In view of the excellent feature extraction performance of the FPN, researchers in the field of machine vision in the past two years have successively proposed models such as path aggregation network (PANet) [37], neural architecture search (NAS-FPN) [38], and bidirectional feature pyramid network (BiFPN) [39, 40] and applied them to image recognition, detection, and segmentation of various scenarios based on their research applications. We proposed the Side Fusion FPN, the main idea of which is to make full use of feature semantic information on feature fusion and feature extraction while increasing the amount of calculation as little as possible. The aim is to take full advantage of the feature map of the information with high semantics, that is, P2 to P6 mentioned above, and combine them side by side.

As shown in Figure 5, as a pioneering method for feature extraction, the FPN proposes a top-down and side-to-side connection method to combine multiscale features. Following this idea, PANet is proposed, which adds an additional path on the basis of the FPN bottom-up to the aggregating network. This method further combines feature semantic information for better feature extraction. The NAS-FPN uses a neural architecture search to obtain irregular feature network topologies. This method can cross-range fusion characteristics and adopt the neural network search technology to form a new feature pyramid structure. Although the NAS-FPN can achieve better performance, it requires thousands of GPU hours in the search process, and the generated feature network is only one thing, so it is difficult to explain. The next method to emerge was BiFPN. It uses two-way cross-scale connection and weighted feature fusion to improve the detection accuracy, but compared with the FPN, it still requires a good deal of calculation.

Again, this is why we proposed the SF-FPN algorithm. It is based on the FPN algorithm, but without increasing any output, it reduced the amount of calculation as much as possible, making full use of high semantic feature information, adding 6 fusions lines, and making the fusion between P2 and P5 and simultaneously making P6 the final fusion output. We also proposed a fully connected FPN as a comparison, that is, on the basis of PANet, all the semantic feature information will have a through-connection.

From the classic FPN structure, compared to C2–C5, we know that P2–P6 have rich feature semantic information. It is cost-effective to fuse these five feature maps. Therefore, this paper designs the Side Fusion FPN in such a way that we add only six side fusion curves on the basis of the FPN. Curve 1: transfer P5 feature semantic information to P3. Curve 2: transfer P5 feature semantic information to P2. Curve 3: transfer P4 feature semantic information to P2.

Table 3: Network structure contrast of Resnet-50/Resnet-86/Resnet-101.

Backbone	Output size	Resnet-50	Resnet-86	Resnet-101
Conv_1	512 ∗ 512		7 ∗ 7, 64, stride2	
Conv_2	256 ∗ 256		3 ∗ 3 maxpool, stride2	
Conv_3	128 ∗ 128	$\left.\begin{matrix}1*1,64\\3*3,64\\1*1,256\end{matrix}\right\}*3$	$\left.\begin{matrix}1*1,64\\3*3,64\\1*1,256\end{matrix}\right\}*3$	$\left.\begin{matrix}1*1,64\\3*3,64\\1*1,256\end{matrix}\right\}*3$
Conv_4	64 ∗ 64	$\left.\begin{matrix}1*1,128\\3*3,128\\1*1,512\end{matrix}\right\}*4$	$\left.\begin{matrix}1*1,128\\3*3,128\\1*1,512\end{matrix}\right\}*4$	$\left.\begin{matrix}1*1,128\\3*3,128\\1*1,512\end{matrix}\right\}*4$
Conv_5	32 ∗ 32	$\left.\begin{matrix}1*1,512\\3*3,512\\1*1,2048\end{matrix}\right\}*3$	$\left.\begin{matrix}1*1,512\\3*3,512\\1*1,2048\end{matrix}\right\}*3$	$\left.\begin{matrix}1*1,512\\3*3,512\\1*1,2048\end{matrix}\right\}*3$
	1 ∗ 1		Average pool, 1000-d fc, softmax	

Table 4: Experimental data.

Backbone_class	FPN + resnet101_81	FPN + resnet86_81	FPN + resnet50_81	FPN + resnet101_3	FPN + resnet86_3	FPN + resnet50_3
Epoch_steps	160_1000	160_1000	160_1000	160_1000	**160_1000**	160_1000
Total params	64158584	58549624	45088120	63744170	**58135210**	44673706
Trainable params	64047096	58453496	45028856	63632682	**58039082**	44614442
FLOPs	130205828	118834293	91542609	129377924	**118006389**	90714705
Memory_size	257.6 M	235.0 M	180.9 M	255.9 M	**233.3 M**	179.2 M
Train_time	27.98 h	21.77 h	18.25	23.02 h	**20.43 h**	18.73 h
Test_avg_time (M4,952)	2.14 s	2.014 s	1.39 s	2.10 s	**1.97 s**	1.36 s

The first row is the network structure and the number of identification categories. For example, FPN + resnet101_81 uses the Resnet-101 residual network to identify 81 types in the FPN. The second row indicates that all network structures are trained 160 ∗ 1000 = 160,000 times. Memory_size refers to the weight of memory after each network structure is trained. Train_time refers to the time taken for each network structure training. Total params and Trainable params represent the total memory parameters and training memory parameters, respectively, of the network structure. Floating point operations (FLOPs) indicate the number of floating-point operations for each network structure, that is, the amount of calculation. Test_avg_time (M4952) refers to the average time to test 4952 images for each network structure. Min_train_loss refers to the minimum value of the weight loss after 160,000 training for each network structure.

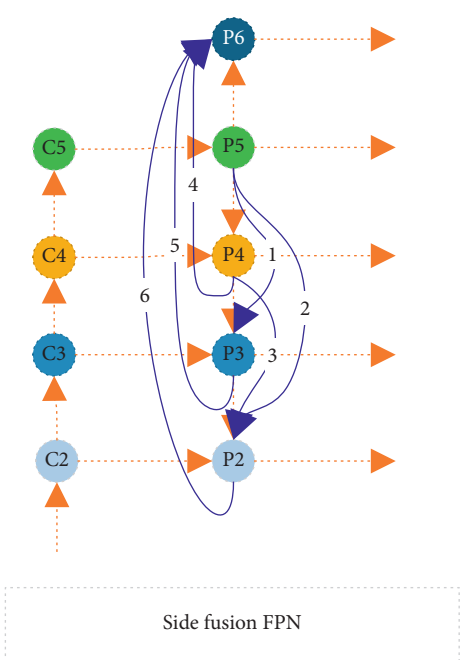


Figure 4: The Side Fusion FPN algorithm structure chart.

Curve 4: transfer P4 feature semantic information to P6. Curve 5: transfer P3 feature semantic information to P6. Curve 6: transfer P2 feature semantic information to P6. After adding the three feature fusion curves P1, P2, and P3, we can see that P2 and P3 can fully converge feature semantic information. At the same time, we have modified it in P6. In the original FPN structure, P6 is directly downsampled by P5, the semantic information only comes from P5, and some of the semantic information is lost during the downsampling process from P5 to P6. We output P6 as the last scale and aggregate the rich feature semantic information of the four scales P2, P3, P4, and P5 at the same time.

This article uses the Side Fusion FPN we proposed as the first part of Mask R-CNN and uses our design of the deep residual network Resnet-86 as the backbone to obtain five scale feature maps. As can be seen from Tables 5 and 6, with the SF-FPN algorithm we designed in the entire network framework, the amount of calculation has only increased by 2.54×10^{-7}. Although this calculation amount is almost minimal, in the test results, mAP has increased by 2.77 points, so there is an obvious improvement in accuracy.

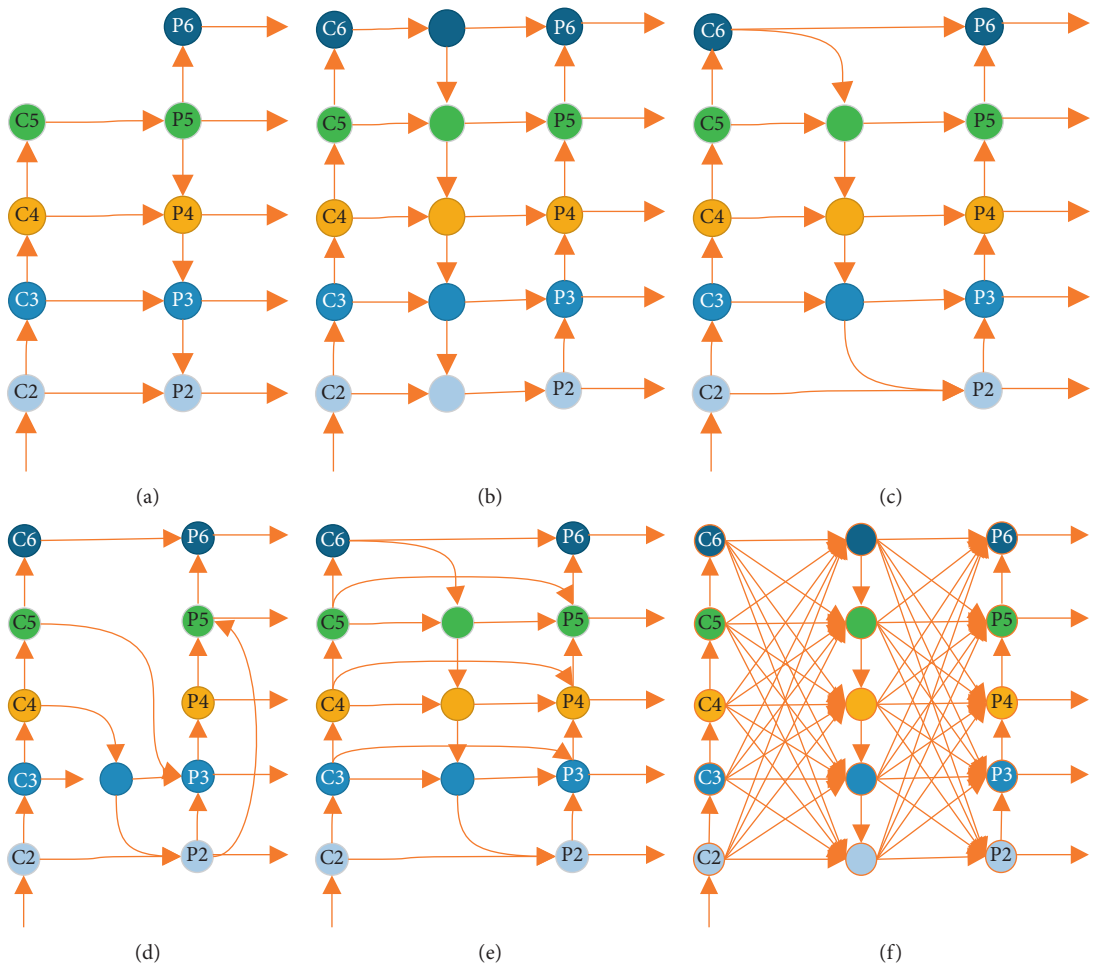


Figure 5: Six kinds of network structure diagrams all improved based on the FPN. (a) FPN. (b) PANet. (c) Simplified PANet. (d) NAS-FPN. (e) BiFPN. (f) Fully connected FPN.

Table 5: Experimental data.

Backbone_class	FPN + resnet101_81	FPN + resnet86_81	FPN + resnet50_81	FPN + resnet101_3	FPN + resnet86_3	FPN + resnet50_3
$mAP^{\mathrm{IOU}=0.5}$	59.11	58.74	46.16	75.69	**74.84**	67.09
$mAP^{\mathrm{IOU}=0.6}$	55.02	54.88	42.04	70.24	**69.23**	62.15
$mAP^{\mathrm{IOU}=0.7}$	47.45	47.88	35.13	60.34	**58.77**	51.98
$mAP^{\mathrm{IOU}=0.8}$	33.35	33.88	22.85	39.96	**39.04**	31.59
$mAP^{\mathrm{IOU}=0.5}_{\mathrm{person}}$	76.76	75.96	70.60	80.10	**78.89**	73.21
$mAP^{\mathrm{IOU}=0.5}_{\mathrm{car}}$	53.29	50.87	45.77	52.73	**51.38**	46.99
$mAP^{\mathrm{IOU}=0.5}_{\mathrm{bus}}$	83.15	81.26	75.86	84.09	**83.28**	76.71
Min_train_loss	0.9564	0.9092	1.287	0.6592	**0.7138**	0.8643

The evaluation standard uses the mAP. The number in the upper right corner of mAP indicates the number of input/output units (IOUs). The words "person," "car," and "bus" in the bottom right corner mean the detection of the single category. Min_train_loss refers to the loss value of each model after training.

3.3. Regional Proposal Network (RPN). The five sets of feature maps generated by the FPN are sent to the RPN. As can be seen from Figure 6, the RPN uses a small grid to slide across five sets of feature maps to produce 758664 boxes and form regional recommendations. We introduce the algorithm for generating candidate frames for RPN. As shown in Figure 6, suppose there are n feature maps in total, the width of each feature map is P_{W_i}, and the feature map is P_{H_i}, we can obtain that the feature maps have $P_{W_i} \bullet P_{H_i}$ pixels. Using each pixel as an anchor point, three scales and three scale candidate frames are generated at the same time, that is, nine candidate frames are generated for each pixel. In this way, we can obtain the number of candidate frames for each feature map as $9 \bullet (P_{W_i} \bullet P_{H_i})$. Therefore, RPN generates the total number of candidate frames after passing all the feature maps.

$$\mathrm{TotalBoxes} = \sum_{i=1}^{n} \left[9 \bullet \left(P_{W_i} \bullet P_{H_i} \right) \right]. \tag{1}$$

Table 6: Experimental data with the SF-FPN algorithm.

Backbone_class	SF-FPN + resnet101_81	SF-FPN + resnet86_81	SF-FPN + resnet50_81	SF-FPN + resnet101_3	SF-FPN + resnet86_3	SF-FPN + resnet50_3
Epoch_steps	160_1000	160_1000	160_1000	160_1000	**160_1000**	160_1000
Total params	64158584	58549624	45088120	63744170	**58135210**	44673706
FLOPs	130205858	118834323	91542639	129377954	**118006419**	90714735
$mAP^{IOU=0.5}$	62.64	58.89	46.69	76.82	**76.64**	69.85
$mAP^{IOU=0.6}$	57.32	55.18	42.79	72.93	**71.88**	64.92
$mAP^{IOU=0.7}$	49.86	48.16	36.03	63.93	**62.41**	54.75
$mAP^{IOU=0.8}$	36.02	34.02	23.16	44.55	**42.04**	34.38
$mAP^{IOU=0.5}_{person}$	79.62	79.20	73.05	82.76	**82.46**	75.21
$mAP^{IOU=0.5}_{car}$	55.82	46.96	48.00	53.58	**52.18**	47.99

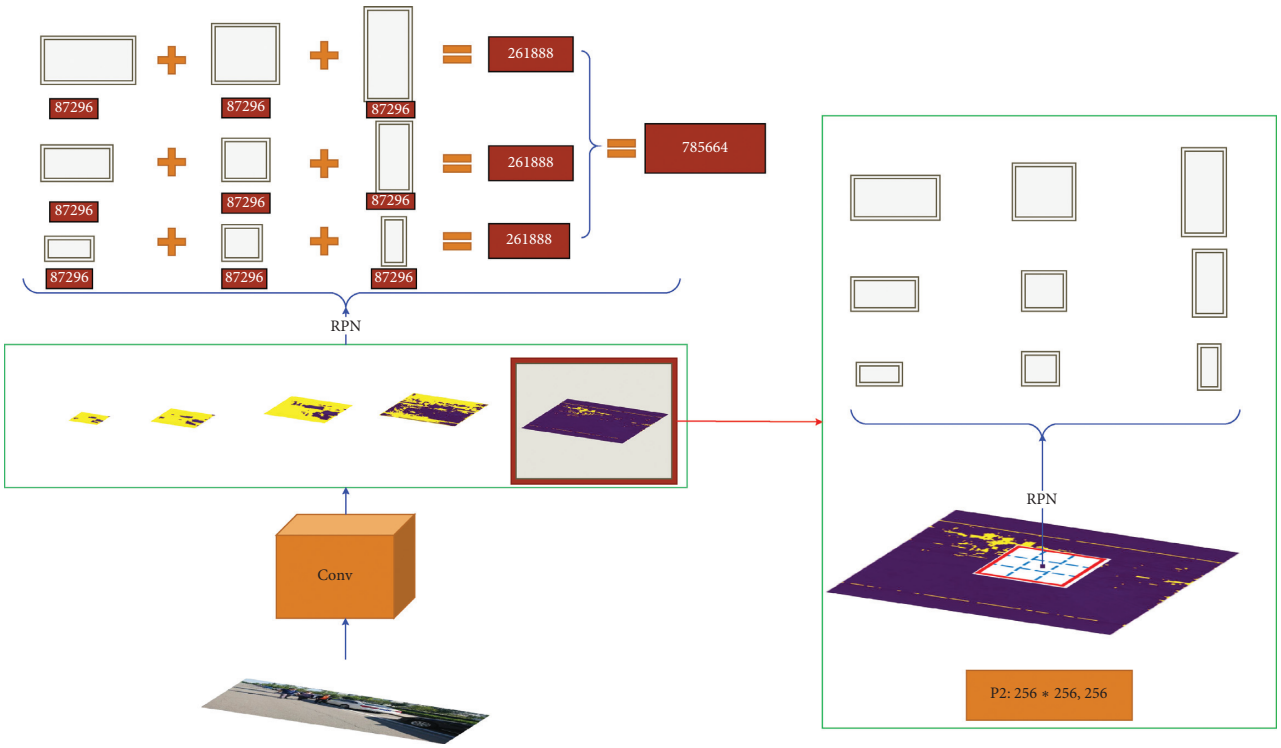


Figure 6: The RPN operation uses each pixel as an anchor point on each feature map and simultaneously generates candidate frames of three sizes and three ratios. All candidate frames are then subjected to NMS screening according to the score, and a certain number of candidate frames are selected and saved, which are used for subsequent training and prediction.

From the above, we can know that this paper generates five sets of feature maps. The size of these five feature maps is $16 * 16$, $32 * 32$, $64 * 64$, $128 * 128$, and $256 * 256$. Through the above formula, we can see that the phases of RPN can generate 758,664 boxes.

Next, the network will calculate 758,664 boxes through nonmaximum suppression (NMS) [38]. The network sorts the scores from large to small according to the four factors of color, texture, total area after merging, and the total area of the merged box in its bounding box and retains 2000 train boxes and 1000 inference boxes. Through the NMS algorithm, a large amount of calculation is required when the network selects to reserve 3000 boxes, and the network performs subsequent training and prediction on the selected 3000 boxes, with a large amount of calculation required. This is therefore the most time-consuming part of the RPN. The 3000 boxes are used to detect 81 types of targets in the COCO dataset, but there are only three categories of detection targets in this study. To increase the speed of the network without affecting the detection accuracy of the target, we need to retain 500 train boxes for training and 250 inference boxes as predictions.

3.4. Dataset Improvement. The detection objects of this paper are cars, buses, and pedestrians. The Microsoft COCO public dataset contains 81 categories, and it contains 82,081 image samples. The parameters in mask_rcnn_COCO.h5 obtained by training the dataset through Mask R-CNN are for detecting 81 kinds of targets. Using this weight directly to detect vehicles and pedestrians can make the calculations too complicated. Therefore, two changes have been made to the COCO dataset: for the first change, we screened the three

categories of images in the COCO dataset (car, bus, and person) and formed a new dataset, which we named the COCO_pcb dataset. For the second change, we labeled the new dataset with an open-source image annotation tool called VGG Image Annotator (VIA). The annotated file is named via_pcb_data.json. The dataset uses 1000 images as the training set, 100 images as the verification set, and 50 images as the test set. VIA was developed by the Visual Geometry Group, and it can be used online or offline. As shown in Figure 7, we can label the target using annotation methods for rectangles, circles, ellipses, polygons, points, and lines. This way, we not only make the sample a valid sample of the three categories of car, bus, and person but also ensure that the number of datasets is sufficient. This will assist in the precise training of the experimental part and improve the accuracy of the target detection.

4. Experiment and Results Analysis

The experimental configuration environment consisted of the Ubuntu 16.04 operating system, Intel Xeon E5 2678 V3 processor (∗2), Nvidia GTX1080TI 11G display card (∗4), Samsung 850 EVO1 TB solid state disk plus 4 TB hard disk, E5 radiator (∗2), RECC DDR4 2133 with 64 GB memory, and QuickPath Interconnect (QPI) 9.6 GT/s motherboard. The operational environment consists of the Python 3 language, TensorFlow 1.3 open-source software library, and Keras 2.0.8 neural network API modeling system.

We used the VIA image annotation tool to polygonize the original image and then sent the newly obtained annotation file via_pcb_data.json to Mask R-CNN network for 160,000 iterations. We named the weight as mask_rcnn_via.h5. Finally, we used mask_rcnn_via.h5 as the initial weight for migration learning. We used the coco_pcb dataset, which has only three categories of car, bus, and person, to perform 160,000 iterations of training with FPN + Resnet101, FPN + Resnet50, and our designed FPN + Resnet86. Under the same environment, 1,000 iterations were performed for SF-FPN + Resnet101, SF-FPN + Resnet50, and SF-FPN + Resnet86, respectively.

This experiment was conducted with FPN + Resnet101, FPN + Resnet86, FPN + Resnet50, SF-FPN + Resnet101, SF-FPN + Resnet86, and SF-FPN + Resnet50 as backbones. We trained it on a COCO dataset containing 81 classes and a COCO_pcb dataset containing three classes. In addition, we used the SF-FPN designed by the author of this article as a backbone to train the COCO_pcb dataset containing only three classes. The epoch for the above 12 sets of experiments is set to 160. Each epoch contains 1000 iterations, and the total number of iterations is 1,920,000. During the experiment, we recorded the memory size of the weights, the total time spent on training, the average length of time on test 4952 images, and the value of mAP on test 4952 images. We also tested the params and FLOP values of 12 groups of experiments. As shown in Figures 8 and 9, we call the tensor board function to draw loss image. Finally, as shown in Figures 8 and 9, we selected the smallest loss value for each set of experiments and recorded it in Tables 4 and 5. We tested 4952 images with the best weights in each set of experiments and calculated the detection time for each network.

As can be seen from Table 4, total params is the first parameter comparison for the six network structures. Params refers to the amount of parameters included in the network structure. For example, for each layer of the n-layer convolutional neural network, the convolution the kernel width is C_{W_i}, kernel length is C_{H_i}, the number of input channels is C_{IN_i}, and the number of output channels is C_{OUT_i}. Then, we can get the memory parameters of each layer of the convolutional neural network:

$$P_{C_i} = \left(C_{W_i} \bullet C_{H_i} \bullet C_{\mathrm{IN}_i} + C_{\mathrm{OUT}_i}\right). \tag{2}$$

Usually, a complete neural network structure also includes the last fully connected layer. We assume that the input of the fully connected layer is N_{IN}, and the output is N_{OUT}. The memory parameters of this fully connected layer are

$$P_N = \left(N_{\mathrm{IN}} \bullet N_{\mathrm{OUT}} + N_{\mathrm{OUT}}\right). \tag{3}$$

Hence, we can get the number of total params:

$$\mathrm{TotalParams} = \left(\sum_{i=1}^{n} P_{C_i}\right) + P_N = \left[\sum_{i=1}^{n}\left(C_{W_i} \bullet C_{H_i} \bullet C_{\mathrm{IN}_i} + C_{\mathrm{OUT}_i}\right)\right] + \left(N_{\mathrm{IN}} \bullet N_{\mathrm{OUT}} + N_{\mathrm{OUT}}\right). \tag{4}$$

4.1. Experimental Data Analysis of FPN + Resnet-50, FPN + Resnet-86, and FPN + Resnet-101. As can be seen from Table 4, compared to Resnet101_81, the params value of Resnet86_3 is reduced by 6,023,374, and the FLOP value is decreased by 12,199,439, which saved nearly 7.55 hours of training time. When testing 4,952 images, the average detection time was reduced by 0.17 seconds, and its detection speed was increased by 7.94%. The speed of Resnet86_3 in training and detection was improved, and its weight memory was reduced by 24.3 MB, which contributes to its implementation in embedded development. As can be seen from Table 5, the minimum loss value of Resnet86_3 is 0.7138, which is 15.73 points higher than the mAP value of Resnet101_81, and it also has a certain improvement in accuracy.

Compared with Resnet86_81, Resnet86_3's params value is reduced by 414,414, and the FLOP value is reduced by 827,904, which shortens the training time by 1.35 hours. When testing 4,952 images, the average detection time is shortened by 0.04 seconds. In terms of loss, it is 0.1954 points lower than the loss value of Resnet86_81, and the mAP value is 16.4 points higher. By comparison, the two structures of Resnet-86_3 and Resnet-86_81 reflect the impact of the number of detection categories in the network.

Compared with Resnet-50_81, although Resnet-86_3 does not display an advantage in params, FLOPs, and speed, we can see from Figure 8 that it is not suited for our purpose. This is because the training effect of Resnet50_81 shows that the smallest loss is high (1.287) and mAP (28.68) is lower than the one in Resnet86_3. This does not meet our standard for detection accuracy. We can also see in Figure 8 that the

Figure 7: Comparison of the rectangular and polygonal marking method: the upper line is the rectangular dimension and the lower line is the polygon dimension.

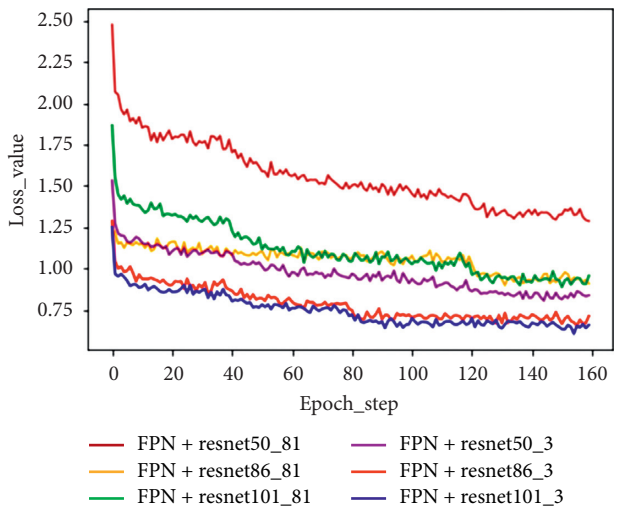


Figure 8: Six experimental loss function graphs with the FPN algorithm.

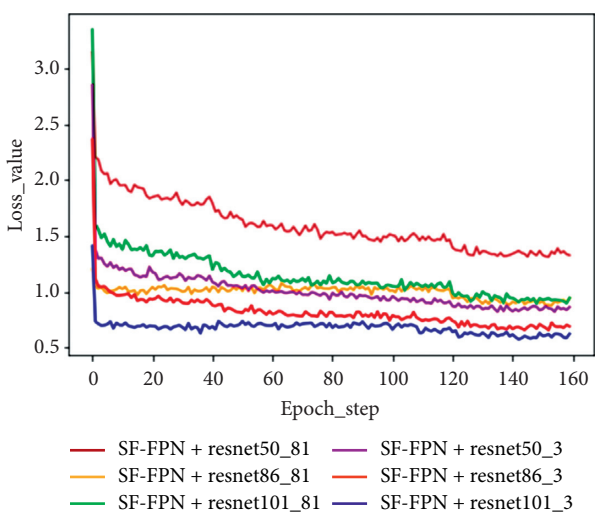


Figure 9: Six experimental loss function graphs with the SF-FPN algorithm.

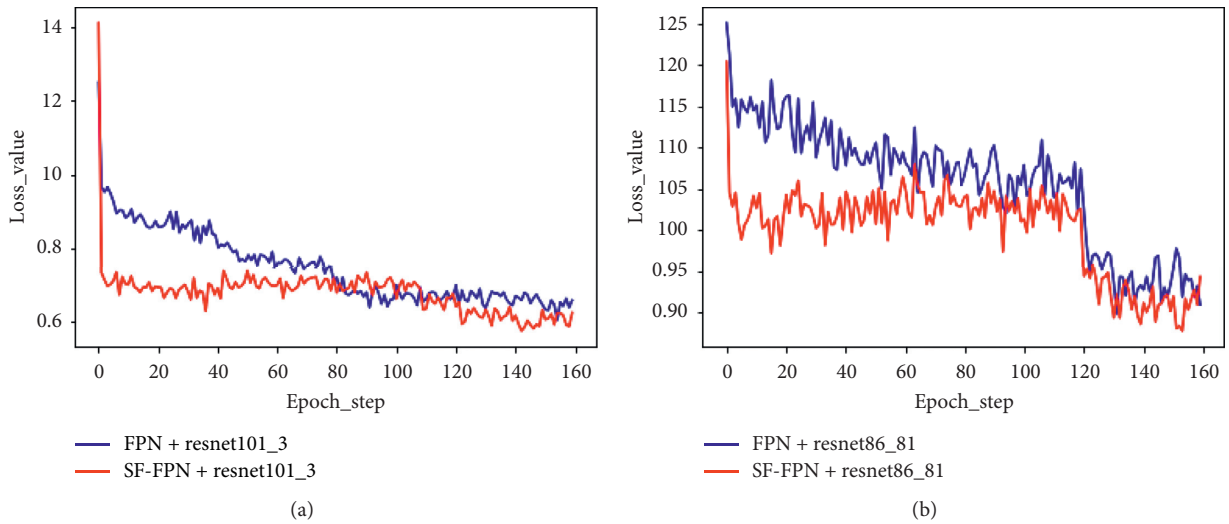


Figure 10: For two algorithms of FPN and SF-FPN, two groups of loss function curves are selected for comparison. (a) Loss function curve of FPN + resnet101_3 and SF-FPN + resnet101_3. (b) Loss function curve of FPN + resnet86_81 and SF-FPN + resnet86_81.

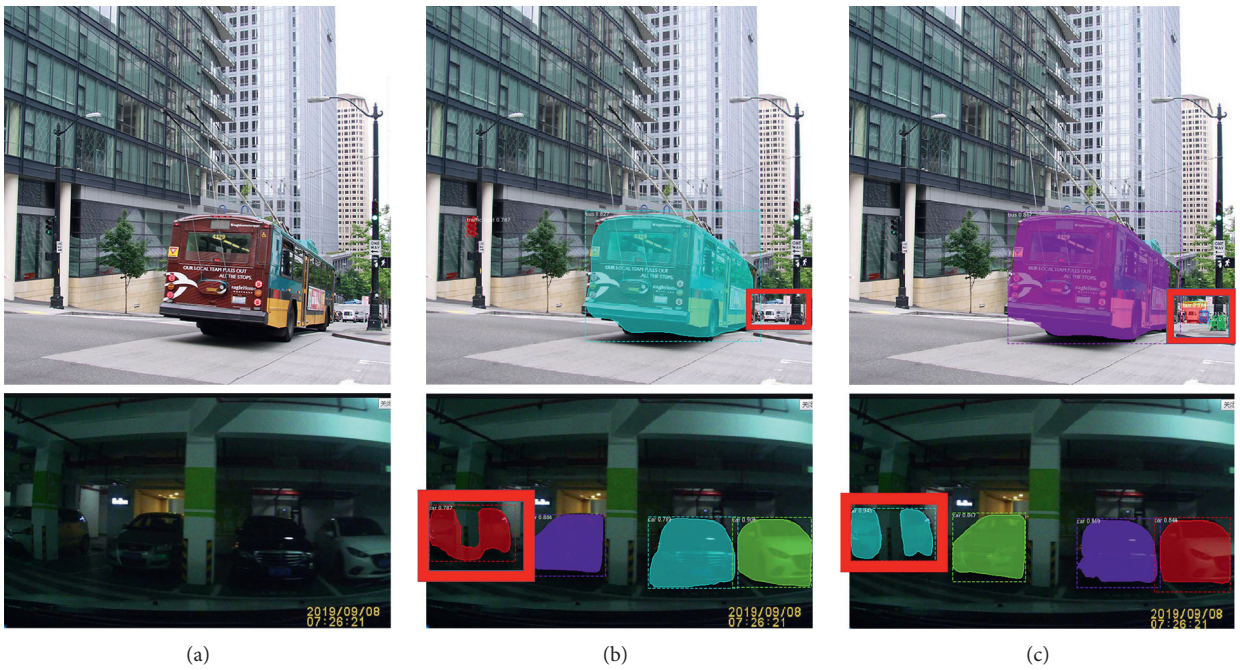
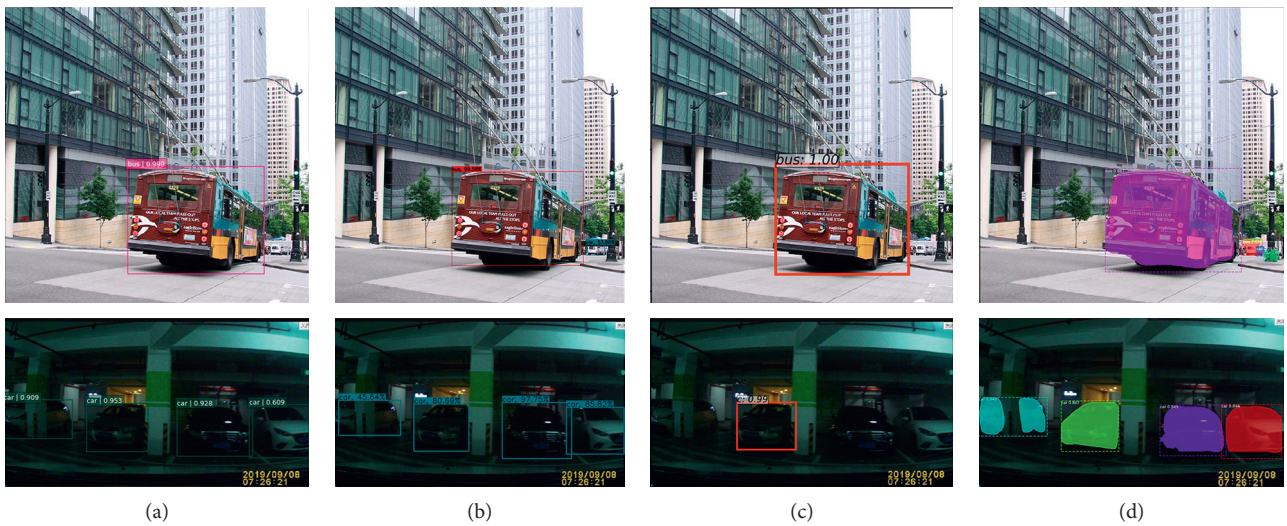


Figure 11: Comparison between the original algorithm and the experimental test result of the algorithm proposed in this study. (a) Original image. (b) Effect map after processing by original algorithm. (c) Effect map after processing by our algorithm.

smallest loss of Resnet50_3 is very high (0.8643) and mAP (7.75) is lower than that of Resnet86_3.

Compared with Resnet101_3, Resnet86_3's params are reduced by 5,608,960, and FLOPs are reduced by 11,371,535, which saves 2.58 hours in training time. When testing 2,895 images, the average test time per image was reduced by 0.13 seconds, and the speed of the test increased by 6.19%. The weight memory of Resnet86_3 was also 22.6 M smaller than the one of Resnet101_3, and the value of mAP was only 0.85 lower than that of Resnet101_3.

It can be seen from Figure 8 that when the Resnet-86 is designed as a backbone in the Mask R-CNN to detect the three major categories of car, bus, and person, its training effect is significantly better than the original Mask with Resnet-101 as the backbone R-CNN. Although the mAP value of Resnet-86_3 is slightly lower than the mAP value of Resnet-101_3, the detection speed of the former one is faster, which meets the goal of our design. Therefore, in the end, we used Resnet-86_3 as the backbone of Mask R-CNN and applied it to the target detection algorithm.



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

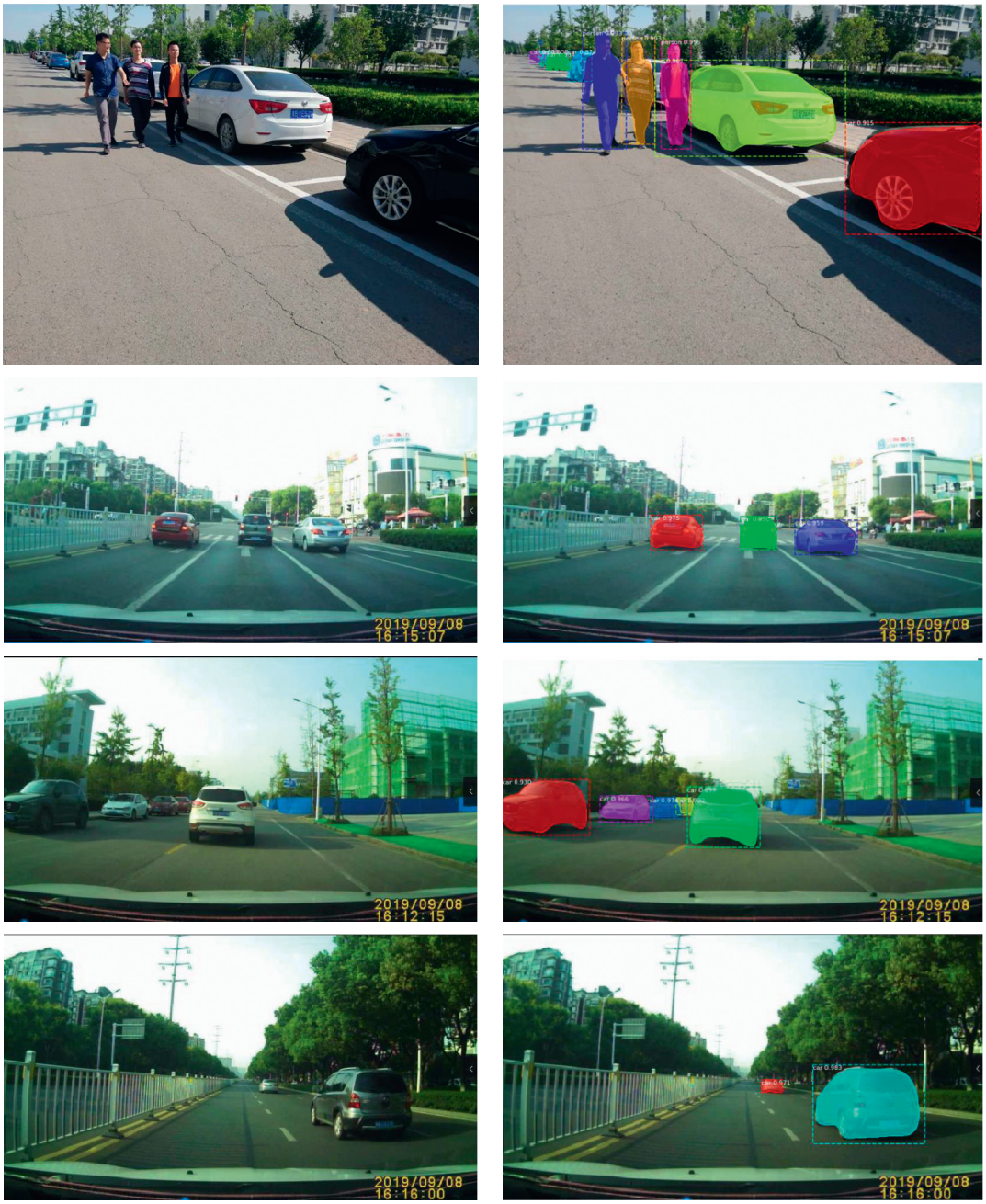
Figure 12: Comparison with state-of-the-art vehicle and pedestrian detection methods. (a) SSD. (b) YOLOv3. (c) Faster R-CNN. (d) Improved Mask R-CNN.

4.2. Analysis of FPN and SF-FPN Experimental Data. By comparing the experimental data of 12 groups in Table 4 and 6, we can see that the params value of the SF-FPN network structure designed in this paper remains unchanged without adding any memory parameters compared with the FPN. As for the SF-FPN, the integration of six side connections has been added, so the six network structures designed based on the SF-FPN algorithm have increased by 30 compared with the original structure FLOPs. Compared with the FPN + resnet101_81 network structure, the total increase is only 2.54×10^{-7}. By testing 4952 images, we can also see that the speed of the test is basically the same as the original speed. However, the mAP value has increased by 2.00. The SF-FPN algorithm, combined with resnet-101 to test 81 classes, leads to a mAP increase of 2.73 points. The SF-FPN algorithm, combined with resnet86 to test 81 classes, leads to a mAP increase of 2.18 points. The SF-FPN algorithm, combined with resnet50 to test 81 classes, leads to a mAP increase of 0.62 points. The SF-FPN algorithm, combined with resnet101 to test three categories, leads to a mAP increase of 2.93 points. The SF-FPN algorithm, combined with resnet-86 to test three categories, leads to a mAP increase of 2.77 points. The SF-FPN algorithm, combined with resnet-50 to test three categories, leads to a mAP increase of 0.75 points.

A comparison of the loss curves for the two groups of experiments selected from Figures 8 and 9 is shown in Figure 10. When the parameters such as the dataset, the number of network layers, the number of detection categories, the training rate, and the number of training iterations during training are the same, we find that the effect is obviously better than the FPN, and the loss value of the SF-FPN's loss curve has always been lower than the FPN's loss value. Particularly in the early stage, it can quickly reach a lower loss value, which has basically been ahead of the FPN training effect.

Taking into account the quality characteristics of the SF-FPN, we finally adopted the SF-FPN as the feature extraction structure, in which the SF-FPN uses Resnet86 as the network structure for object recognition, detection, and classification of the three categories of car, bus, and person. Compared with the original FPN + Resnet101 structure recognition class 81, we designed the network structure SF-FPN + Resnet86_3 to improve the training speed by 26.98% and improve the mAP accuracy evaluation by 17.53 points. As shown in Figure 11, we tested two images with the above two algorithms, respectively, in which the FPN + resnet101_81 algorithm missed the small target vehicle in the red frame in the first image, while the SF-FPN + resnet86_3 algorithm proposed in this paper accurately detected the small target vehicle in the red frame. When detecting the second image, the FPN + resnet101_81 algorithm does not segment the red table frame accurately in the segmentation process, while our algorithm SF-FPN + resnet86_3 segments the part accurately, distinguishing between the vehicle and nonvehicle parts. At the same time, the network structure framework we designed can be easily migrated to other network structure models, such as Faster R-CNN, SSD, and YOLOv3.

At the end of this section, we would like to provide a short comparative remark on our method and some state-of-art methods. After testing, the mAP value of SSD-based vehicle detection algorithm [41] is only 50.4%, which is 26.24% lower than our new algorithm. As shown in Figure 12(a), in the detection of the first test diagram, the target at the lower right corner is not detected, and there is a missed detection. The mAP value of vehicle detection algorithm [42] based on YOLOv3 is 57.9%, which is 18.74% lower than the new algorithm. It can be seen from Figure 12(b) that the detection and positioning of the leftmost target "car" in the second image is not accurate enough. The mAP value of Faster R-CNN vehicle detection algorithm [43] is 59.1%, but it is still 17.54% lower than the new algorithm. It can be seen from Figure 12(c) that there is a missed inspection in the vehicle detection of both figures. At the same time, in conjunction with the *cascaded superpixel pedestrian object segmentation*



(a) (b)

Figure 13: The four images on the left are the original images, and the four images on the right are the experimental results. The four results are obtained by the network algorithm designed in this paper, which improves on the FPN, RPN, and categories. (a) Road original image. (b) Mask R-CNN test chart.

algorithm [44], *bibox regression for pedestrian detection algorithm* [45] and other algorithms are compared, and the experimental results show that the vehicle and pedestrian detection based on the improved Mask R-CNN is slightly more accurate in the task of case segmentation.

5. Conclusions

The main research content of this study is about how to make the Mask-RCNN algorithm detect and segment cars, buses, and persons on the road more accurately and more quickly in the anticollision warning system. To increase the accuracy of the training effect of the network, we filtered and supplemented the dataset. To meet the real-time requirements of smart driving, we designed the Resnet-86 network and used it as a network backbone. To further increase the detection speed of the network, we modified the number of reserved RPN candidate frames. For greater accuracy, we designed the SF-FPN algorithm for feature extraction. Through improving the dataset, FPN, and RPN, our network

improved the detection speed by 7.94%, and the detection accuracy increased the value of mAP by 17.53 points over the original Mask R-CNN network.

Based on Mask R-CNN, we improved the network to integrate the functions of image recognition, detection, and segmentation. As we can see from Figure 13, the improved network can accurately detect a distance of about 200 m, even though the target is occluded by 95%. It can be seen that the network can be applied to the intelligent driving anticollision warning system to identify the car, bus, and person ahead.

Although the recognition speed of this network has reached 5 FPS, for some real-time system applications, the recognition speed still needs to be increased and the hardware configuration requirements need to be reduced. For example, in the vehicle tracking task, the target detection speed needs to be completed faster; in the precision instrument segmentation task, the target needs to be segmented more accurately; in the vehicle emergency braking device, the target needs to be detected faster to complete emergency braking.

In the future, in order to improve the detection accuracy, we can further improve and design the feature extraction algorithm to make the feature semantic information more abundant. It can also further optimize the depth residual network, reduce the loss function value, and improve the network training effect. In order to improve the detection speed, we can combine and optimize the deep convolution neural network algorithm, reduce the network computing redundancy, and improve the network detection speed. It can also be combined with hardware configuration to enhance the network computing ability and further improve the speed of target detection. Therefore, how to improve the speed and accuracy of target detection is still our key research work in the future.

Data Availability

The data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 6 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported in part by the National Natural Science Foundation of China (no. 11971034).

References

[1] J. Nie, "Design of visual feature detection system for intelligent driving of electric vehicle," in *Proceedings of the International Conference on Robots & Intelligent System (ICRIS)*, ACM, Amsterdam, Netherlands, February 2018.

[2] E. Sun, A. Nieto, and Z. Li, "GPS and Google Earth based 3D assisted driving system for trucks in surface mines," *International Journal of Mining Science and Technology*, vol. 20, pp. 138–142, 2016.

[3] Y.-S. Chen, S.-C. Chiu, and S.-S. Hsiau, "Safe technology with a novel rear collision avoidance system of vehicles," *International Journal of Automotive Technology*, vol. 20, no. 4, pp. 693–699, 2019.

[4] H. Kim, J. Kim, and Y. Kim, "Design of network threat detection and classification based on machine learning on cloud computing," *Cluster Computing*, vol. 22, pp. 1–10, 2019.

[5] H. Xing, G. Zhang, and M. Shang, "Deep learning," *International Journal Of Semantic Computing*, vol. 10, pp. 417–439, 2016.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *In*, IEEE, in *Proceedings of the IEEE Computer Society Conference On Computer Vision And Pattern Recognition*, June 2005.

[7] M. M. Adankon and M. Cheriet, "Support vector machine," *Computer Science*.vol. 1, pp. 1–28, 2002.

[8] N. Faris, S. Ali, and M. Z. A. Ilham, "Image-based feature extraction technique for inclined crack quantification using pulsed eddy current," *Chinese Journal of Mechanical Engineering*, vol. 32, pp. 19–34, 2019.

[9] K. Phil, *MATLAB Deep Learning*, pp. 103–121, Berkeley, New York, NY, USA, 2017.

[10] Y. L. Cun, B. Boser, J. S. Denker, D. Henderson, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in Neural Information Processing Systems*, vol. 2, pp. 396–404, 1997.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the International Conference On Neural Information Processing Systems*, IEEE, Lake Tahoe, NV, USA, December 2012.

[12] R. Girshick, J. Donahue, T. Darrel, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the CVPR*, IEEE, Columbus, Ohio, USA, June 2014.

[13] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference On Computer Vision*, IEEE, Santiago, Chile, December 2015.

[14] S. Ren, K. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[15] K. M. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference On Computer Vision*, IEEE, Venice, Italy, October 2017.

[16] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: common objects in context," in *Proceedings of the European Conference On Computer Vision*, Springer, Zurich, Switzerland, September 2014.

[17] V. Bianco, P. L. Mazzeo, M. Paturzo, C. Distante, and P. Ferraro, "Deep learning assisted portable IR active imaging sensor spots and identifies live humans through fire," *Optics and Lasers in Engineering*, vol. 124, 2020.

[18] P. L. Mazzeo, A. Argentieri, F. D. Luca et al., "Convolutional neural networks for recognition and segmentation of aluminum profiles," *Multimodal Sensing: Technologies and Applications*, vol. 11059, 2019.

[19] Q. C. Pan and H. H Zhang, "Key algorithms of video target detection and recognition in intelligent transportation systems," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 4, pp. 142–162, 2019.

[20] D. W. Du, Y. K. Qi, and H. Y. Yu, "The unmanned aerial vehicle benchmark: object detection and tracking," in

Proceedings of the European Conference On Computer Vision, Springer, Munich, Germany, September 2018.

[21] W. Ge, S. Yang, and Y. Yu, "Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning," in *Proceedings of the CVPR*, IEEE, Salt lake city, UT, USA, August 2018.

[22] S. Xi, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the CVPR*, IEEE, Honolulu, HI, USA, July 2017.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference On Computer Vision & Pattern Recognition*, IEEE, Las Vegas, NV, USA, June 2016.

[24] T. Y. Lin, P. Dollár, R. Girshick, K. He, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the CVPR*, IEEE, Honolulu, HI, USA, July 2017.

[25] A. K. Fattal, M. Karg, C. Scharfenberger, and J. Adamy, "Saliency-guided region proposal network for cnn based object detection," in *Proceedings of the IEEE International Conference On Intelligent Transportation Systems*, IEEE, Yokohama, Japan, October 2017.

[26] Z.-b. Zhao, L. Zhang, Y.-c. Qi, and Y.-y. Shi, "A generation method of insulator region proposals based on edge boxes," *Optoelectronics Letters*, vol. 13, no. 6, pp. 466–470, 2017.

[27] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, pp. 640–665, 2014.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proceedings of the European Conference On Computer Vision*, Springer, Zurich, Switzerland, September 2014.

[30] A. Bulat, J. Yang, and G. Tzimiropoulos, "To learn image super-resolution, use a GAN to learn how to do image degradation first," in *Proceedings of the European Conference On Computer Vision*, Springer, Munich, Germany, September 2018.

[31] B. H. Chen, Y. G. Wang, and G. Wei, "End-to-End trained sparse coding network with spatial pyramid pooling for image classification," *Neural Processing Letters*, vol. 1, pp. 1–16, Jan. 2019.

[32] H. Zhang, V. Sindagi, and V. M. Patel, "Multi-scale single image dehazing using perceptual pyramid deep Network," in *Proceedings of the CVPR*, IEEE, Salt lake city, UT, USA, August 2018.

[33] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral-spatial hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 740–754, 2019.

[34] R. Gail, M. Paular, R. Gary, E. Mary, and K. S. Heidi, "Top-down and bottom-up treatment approaches compared in a single case of pure alexia," *Aphasiology*, vol. 1, pp. 1–14, 2019.

[35] J. Chen, Z. Cai, J. Lai, and X. H. Xie, "Efficient segmentation-based patchmatch for large displacement optical flow estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 99, p. 1, 2018.

[36] F. Dornaika, K. Hammoudi, M. Melkemi, and T. D. A. Phan, "An efficient pyramid multi-level image descriptor: application to image-based parking lot monitoring," *Signal Image and Video Processing*, vol. 11, pp. 1–7, 2019.

[37] S. Liu, L. Qi, H. F. Qin, "J. Shi, and J. Jia., "Path aggregation network for instance segmentation," in *Proceedings of the CVPR*, IEEE, Salt lake city, UT, USA, August 2018.

[38] G. Ghiasi, T. Lin, R. Pang, and V. Le, "Nas-fpn: learning scalable feature pyramid architecture for object detection," in *Proceedings of the Conference On Computer Vision And Pattern Recognition (CVPR)*, IEEE, Los Angeles, CL, USA, June 2019.

[39] S. Woo, S. Hwang, H.-D. Jang, and I. S. Kweon, "Gated bi-directional feature pyramid network for accurate one-shot detection," *Machine Vision and Applications*, vol. 30, no. 4, pp. 543–555, 2019.

[40] J. Hosang, R. Benenson, and B. Schiele, "A convnet for non-maximum suppression," in *Proceedings of the German Conference On Pattern Recognition*, Springer International Publishing, Stuttgart, Germany, September 2016.

[41] X. Le, L. Xiang, and Z. Hua, "Vehicle Detection on Road Frontage Based on SSD," *Software Guide*, vol. 5, pp. 27–36, 2019.

[42] F. K. Zhang, Y. Feng, and L. I. Ce, "Fast vehicle detection method based on improved YOLOv3," *Computer Engineering and Applications*, vol. 3, pp. 133–139, 2019.

[43] J. Sang, P. Guo, and Z. Xiang, "Vehicle detection based on faster-RCNN," *Chongqing Daxue Xuebao/Journal of Chongqing University*, vol. 40, pp. 32–36, june. 2017.

[44] D. Yang, J. Huang, J. Zhang, and R. Zhang, "Cascaded superpixel pedestrian object segmentation algorithm," *Control and decision making in China (CCDC)*, vol. 5, pp. 41–48, 2018.

[45] C. Zhou and J. Yuan, "Bi-box regression for pedestrian detection and occlusion estimation," in *Proceedings of the European Conference On Computer Vision*, Springer, Munich, Germany, September 2018.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 3510313, 10 pages
https://doi.org/10.1155/2020/3510313

Research Article

A Weakly Supervised Method for Mud Detection in Ores Based on Deep Active Learning

Zhijian Huang,[1] **Fangmin Li**,[1] **Xidao Luan**,[1] **and Zuowei Cai**[2]

[1]*School of Computer Engineering and Applied Mathematics, Changsha University, Changsha 410003, China*
[2]*School of Information Science and Engineering, Hunan Women's University, Changsha 410004, China*

Correspondence should be addressed to Fangmin Li; lifangmin@whut.edu.cn

Received 1 February 2020; Revised 12 April 2020; Accepted 20 April 2020; Published 30 May 2020

Guest Editor: Chunjia Han

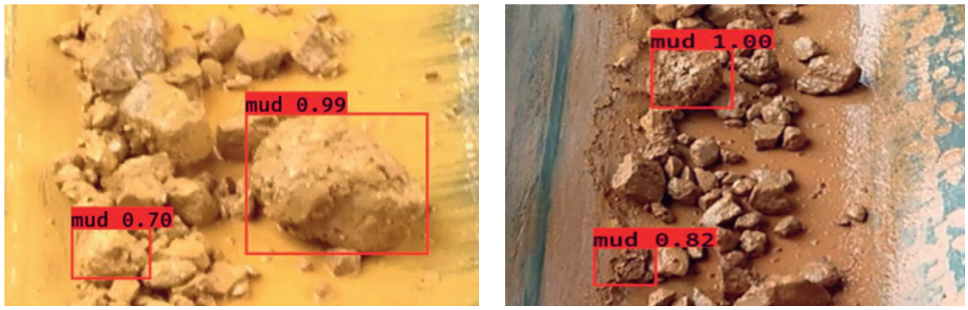
Automatically detecting mud in bauxite ores is important and valuable, with which we can improve productivity and reduce pollution. However, distinguishing mud and ores in a real scene is challenging for their similarity in shape, color, and texture. Moreover, training a deep learning model needs a large amount of exactly labeled samples, which is expensive and time consuming. Aiming at the challenging problem, this paper proposed a novel weakly supervised method based on deep active learning (AL), named YOLO-AL. The method uses the YOLO-v3 model as the basic detector, which is initialized with the pretrained weights on the MS COCO dataset. Then, an AL framework-embedded YOLO-v3 model is constructed. In the AL process, it iteratively fine-tunes the last few layers of the YOLO-v3 model with the most valuable samples, which is selected by a Less Confident (LC) strategy. Experimental results show that the proposed method can effectively detect mud in ores. More importantly, the proposed method can obviously reduce the labeled samples without decreasing the detection accuracy.

1. Introduction

Bauxite is usually mixed with a large amount of mud lump, which is the main impurity in alumina ore. It requires a large dose of chemical reagents (such as alkali) for removal of the mud, which increases the production cost and environmental pollution. More seriously, the mud is highly viscous, which likely blocks production equipment and affects the stability of production. At present, mud removal still relies on traditional manual operating. So, automatically detecting and removing mud from ores with AI technology is important and valuable for production cost reduction and environmental pollution.

However, it is challenging to distinguish the mud and the ore in a real scene. The reason lies in several aspects. (1) Since the mud and the ore are both in the form of lumps, the shape difference is not obvious. (2) Since ore usually cannot be cleaned thoroughly, there is little difference between the mud and the ore in color and texture (see Figure 1). Even experienced experts need careful identification to distinguish. (3) One image often contains multiple pieces of mud whose sizes vary significantly (diameter from 50 mm to 500 mm). (4) More seriously, since the ores from different mines have different compositions and contents, their color and texture have obvious differences.

Currently, there is no special method for mud detection in ores. But we can benefit from the common object detection method which is usually based on deep neural networks and trained with an amount of exactly labeled samples. There are two typical methods: the region-proposal-based method and the regression-based method. The former is also called the two-stage method. A region proposal algorithm finds candidate object regions in the first stage, and then a CNN network extracts the features and classifies candidate objects in the second stage. These methods include the R-CNN [1], the Fast R-CNN [2], the Faster R-CNN [3], the SPP-NET [4], the SSD [5], the R-FCN [6] and the newest Cascaded RCNN [7]. The latter treats object detection as a regression problem and predicts the location and category at the same time. The most representative ones are the YOLO deep neural networks, including the YOLO [8], the YOLO9000 [9], and the YOLO-v3



(a) (b)

Figure 1: The mud in ores. There are only slight differences between mud and ore in shape, color, and texture, and their scales vary significantly. The red squares with confidences are mud.

[10]. Compared with these two typical methods which emerged at the same time, the former is more accurate while the latter is faster overall.

There are several problems in directly using the aforementioned methods for mud detection. Firstly, since the mud and the ore are difficult to distinguish, the common object detection method cannot give a highly accurate result. It needs a special and finer model to give higher accuracy. Secondly, the aforementioned methods are strongly supervised, which need a large number of exactly labeled samples to train a model. Since there is similarity between mud and ores in the real scene, even experienced experts need careful identification to distinguish. So, it is expensive and time consuming to exactly label a large number of samples. Last but not least, since the ores from different mines have different colors and textures, it needs a model that can be transferred easily from a mine to another, which is important for mud detection.

To solve the challenging problem, this paper proposed a weakly supervised method based on deep active learning (AL), named YOLO-AL. The method uses the YOLO-v3 model as the basic detector, which is initialized with the pretrained weights on the MS COCO dataset. Then, an AL framework-embedded YOLO-v3 model is constructed. In the AL framework, it iteratively fine-tunes the last few layers of the YOLO-v3 model with the most important samples.

This paper focuses on the important problem of automatically detecting mud in ores, which is rarely studied. The contributions are summarized in three aspects. (1) We propose a weakly supervised method based on deep active learning for detecting mud in ores, which extensively reduces human labor for annotating training data while achieving performance comparable with the fully supervised learning approaches. (2) We propose a sample selection method based on Less Confident (LC) strategy, which selects the most valuable samples according to the confidences. The confidence of an object is calculated with the scores predicted by the YOLO-v3 detector. (3) Since the proposed method only fine-tunes the last few layers of the YOLO-v3 model with the most valuable samples, it can easily be transferred from one mine to another.

2. Related Work

Active learning [11, 12] assumes that the ground-truth labels of unlabeled instances can be queried from a database [13]. For simplicity, it assumes the labeling cost only depends on the number of queries. Thus, the goal of active learning is to minimize the number of queries. Such that, the labeling cost for training a good model can be minimized. Given a small set of labeled data and abundant unlabeled data, active learning attempts to select the most valuable unlabeled instance to query [13].

Active learning is always used in scenes where data collection is convenient while sample labeling is expensive. Kapoor et al. [14] combined active learning with Gaussian stochastic processes for object categorization. Yang et al. [15] used the AL to train a group of fully convolutional networks (FCN) for biomedical image segmentation. Sun et al. [16] proposed an AL framework based on MRF model for the spectral-spatial classification of hyperspectral imagery. Yang et al. [17] proposed a semisupervised batch mode multiclass active learning algorithm for visual concept recognition, which selects uncertainty sampling with diversity maximization. Dutt Jain and Grauman [18] proposed an active learning method for natural scene image segmentation, which achieves state-of-the-art level performance using significantly less training data.

Recently, weakly supervised learning, in which training sets require only binary labels indicating whether an image contains the object or not, has attracted considerable attention. Han et al. proposed a novel object detection framework by combining the weakly supervised learning and high-level feature learning [19]. Zhou et al. developed a transferred deep model to extract high-level features for object detection from remote sensing images by pretraining a convolutional neural network model on a large-scale annotated dataset and then transferring it by domain-specific fine-tuning [20]. Cheng. et al. trained rotation-invariant and Fisher discrimination CNN models for rotational object detection by imposing a rotation-invariant regularizer and a Fisher discrimination regularizer to the objective function [21]. Cheng. et al. introduced a new rotation-invariant layer on the basis of the existing CNN architectures and learned a

rotation-invariant CNN for object detection from remote sensing image [22].

However, there are few papers that use active learning for object detection, especially for low-distinguishable objects (such as mud and ore). This paper proposed an AL method-integrated YOLO-v3 model for mud detection in ores. The YOLO-v3 model detects the mud and predicts its class confidence and box bound confidence. Based on these confidences, the AL selects the most valuable samples to be labeled. With the gradually increasing labeled samples, a more accurate YOLO-v3 model is trained. The method will bring at least two benefits. (1) Only the most valuable samples are selected to the expert for labeling, which will reduce the number of training samples. (2) Since the expert only needs to check and modify the labels instead of relabeling, the work of labeling is reduced further.

3. Method

3.1. The Overall Framework. The overall framework of the proposed deep active learning method, named the YOLO-AL, is shown in Figure 2, which contains four basic modules: YOLO-v3 model fine-tuning, object detection, sample selection, and expert verification.

As shown in Figure 2, the proposed method is actually an iterative training process. At the beginning, we initiate a YOLO-v3 model with the weight pretrained on the MS COCO dataset. Then, it fine-tunes the last few layers with a little of labeled mud and ore samples to get a new mud detector. With the mud detector, all the unlabeled samples are tested and given confidence to each object. Based on the confidence, a sample selection method selects the most valuable samples which are sent to the expert. The expert checks and modifies the labels of these samples and adds these samples to the labeled training set. With the updated training set, the YOLO-v3 model will be fine-tuned again. The process iterates with gradually increasing labeled samples until reaching the termination condition.

3.2. YOLO-v3 Model Fine-Tuning. The YOLO-v3 model was proposed in [10] for general object detection in the nature scenes. Due to its excellent performance on the speed and accuracy, we use the YOLO-v3 as the detector. We initialize the YOLO-v3 model with the pretrained weights on the MS COCO dataset (http://images.cocodataset.org). Then, we fine-tune the last layers of the YOLO-v3 model iteratively in the AL framework.

As Yosinski et al. [23] pointed out, fine-tuning a deep neural network can preserve the general feature and overcome the difference between datasets to extract special high level features, which help us to quickly construct a new model on a new dataset. In this paper, we froze the Darknet-53 of the YOLO-v3 and fine-tuned the last layers shown in Figure 3. It is worth noting that the Darknet-53 has far more layers and weight parameters than the layers in the dashed box.

In Figure 3, the DBL unit consists of three layers: convolutional (Conv), Batch Normalization (BN), and Leaky ReLU Activation Layer. The ResUnit is a unit with the residual structure. The Resn is composed by a zero-padding, a DBL, and n ResUnit. The DBLU unit consists of a DBL layer and an upsampling layer, while the DBLC consists of a DBL layer and a convolutional layer. The Concat layer combines features at different scales.

The loss function of the YOLO-AL is defined as follows:

$$\begin{aligned} f_{\text{loss}} = {} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{\text{obj}} \Big[\left(b_x - \hat{b}_x\right)^2 + \left(b_y - \hat{b}_y\right)^2 + \left(b_w - \hat{b}_w\right)^2 \\ & + \left(b_h - \hat{b}_h\right)^2 \Big] \\ & + \lambda_{\text{obj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{\text{obj}} \left[-\log\left(p_c\right) + \sum_{k=1}^{n} \text{BCE}\left(\hat{c}_k, c_k\right) \right] \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{\text{noobj}} \left[-\log\left(1 - p_c\right) \right], \end{aligned} \tag{1}$$

where b_x and b_y are the location, while b_w and b_h are the width and the height of the predicted box. S is the number of the grid, namely, S^2 is usually set as 13×13, 26×26, and 52×52 for scales from coarse to fine. B is the predicted box number. $1_{i,j}^{\text{obj}}$ is defined as

$$1_{i,j}^{\text{obj}} = \begin{cases} 1, & \text{if there is a real object in the box,} \\ 0, & \text{else.} \end{cases} \tag{2}$$

and $1_{i,j}^{\text{noobj}}$ is the opposite of $1_{i,j}^{\text{obj}}$ which is defined as

$$1_{i,j}^{\text{noobj}} = \begin{cases} 1, & \text{else,} \\ 0, & \text{if there is a real object in the box.} \end{cases} \tag{3}$$

BCE is binary cross entropy:

$$\text{BCE}\left(\hat{c}_k, c_k\right) = -\hat{c}_k \times \log\left(c_k\right) - \left(1 - \hat{c}_k\right) \times \log\left(1 - c_k\right). \tag{4}$$

p_c is the object class probability. λ_{coord}, λ_{obj}, and the λ_{noobj} are the proportions of the three parties.

3.3. The Sample Selection. The sample selection strategy is the core of the AL [24]. Since the proposed AL method shown in Figure 2 is used for object detection, the sample selection strategy is defined based on the predicted results of the YOLO-v3 detector. It predicts the class probability of each object, based on which we can calculate the confidence. As shown in Figure 4, the YOLO-v3 predicts a vector containing 3 boxes for each grid in the feature map. Each box is a prediction of an object where p_0 is the objectness score and $p_1 \sim p_n$ are the class scores for n-classes. Here, we only consider 3 classes, namely, the mud, the ore, and others. The objectness score p_0 indicates the possibility of whether the box contains an object, namely, $p(object)$, while the class score is the posterior probability $p(\text{class} \mid \text{object})$. So, the confidence of a box can be calculated as follows:

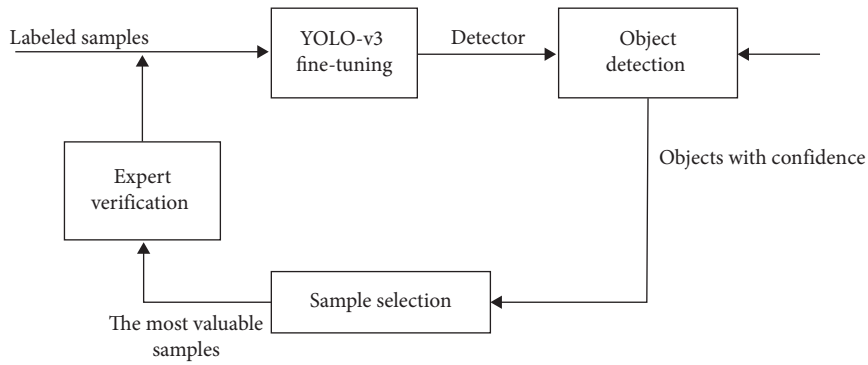


Figure 2: The flowchart of the proposed YOLO-AL method.

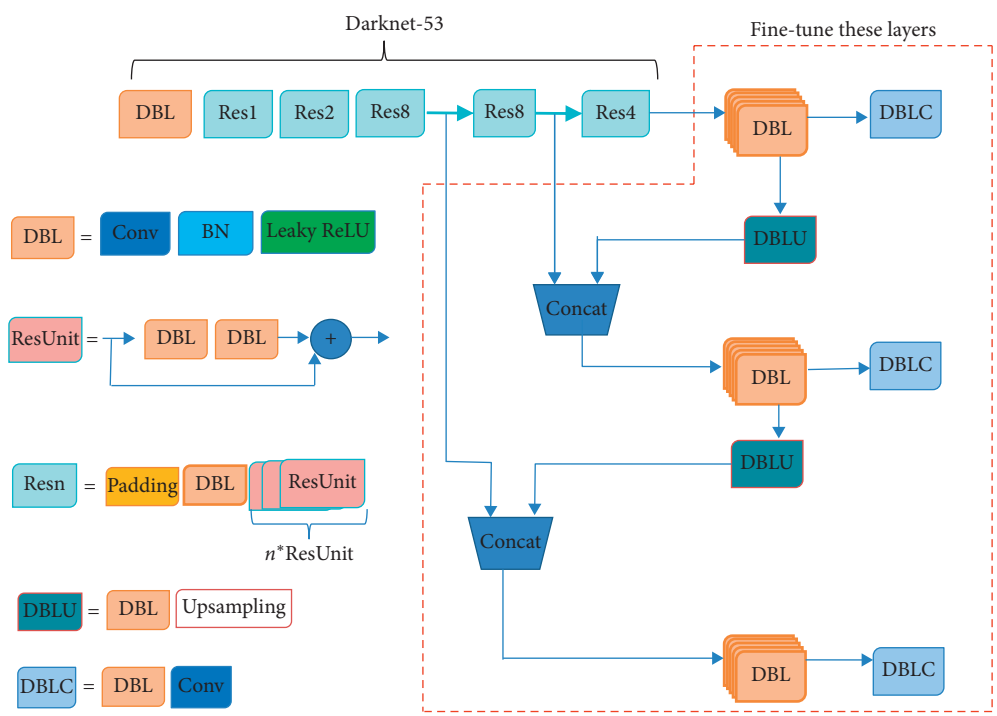


Figure 3: The structure of the YOLO-AL where the Darknet-53 is trained with COCO dataset and the layers in the dashed box are fine-tuned iteratively in an AL process.

$$p(\text{class}) = p(\text{object}) \times p(\text{class} \mid \text{object}). \tag{5}$$

In AL, the strategy of sample selection decides which sample to query or to be labeled by experts. In this paper, we consider two sample selection strategies, the random selection (RS) and the less confident (LC).

The RS method is referred to as passive selection method in contrast to the active selection methods. In the RS method, unlabeled candidates are selected randomly without any active criterion. The RS method is often served as the baseline to be compared with the active selection methods.

The LC method selects the samples with less confident based on the posterior probabilities of all the classes. When using a probabilistic model for binary classification, the LC method selects the sample whose posterior probability is near 0.5.

$$p_{\text{uncertain}} = \left| \max_{m \in L} p(y_i = m \mid \mathbf{x}_i) - 0.5 \right|, \tag{6}$$

$$\hat{\mathbf{x}}_{LC} = \arg\min_{x_i \in D_U} (p_{\text{uncertain}}), \tag{7}$$

where $\max_{m \in L} p(y_i = m \mid \mathbf{x}_i)$ means the most possible label of sample $\mathbf{x}_i$ is m and D_u is the unlabeled dataset. Specific to the problem of mud detection based on YOLO-v3, $p(y_i = m \mid \mathbf{x}_i)$ is actually $p(\text{class})$ in formula (5).

If a candidate object satisfies condition (7), it is considered to be an object containing the most useful information and should be labeled as a training sample. Considering the efficiency of the model training, we sort the mud objects in ascending order according to $p_{\text{uncertain}}$ and take the first n mud objects to be labeled. If there is an

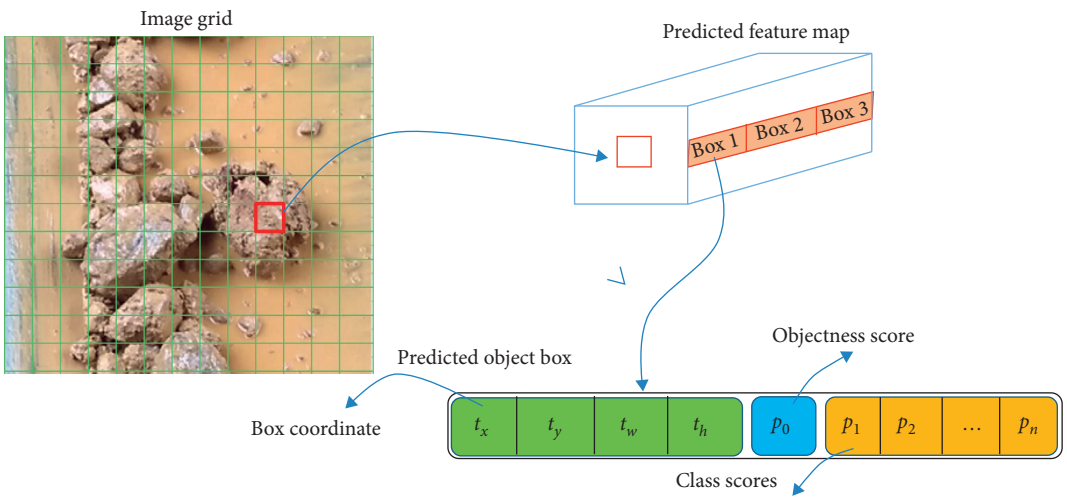


Figure 4: The object with confidence predicted by YOLO-v3. The confidence is the product of the class score and the objectness score.

uncertain object in an image, the image with its labels is put into a dataset $I_{\text{uncertain}}$ which will be recommended to expert to check and modify labels.

Since the images recommended to the expert have predicted objects with labels, the experts only need to modify the class labels or box bounds instead of relabeling them, which will also reduce the work of sample labeling.

3.4. The Algorithm. The YOLO-v3-based AL framework is provided in Algorithm 1.

4. Datasets and Experiment Design

4.1. Dataset Description. The open MS COCO [25] dataset is used to pretrain a YOLO-v3 model. It is downloaded from http://images.cocodataset.org, which contains 12 major classes and 80 subclasses. To compare with the mud and the ore in the Ore Dataset, we only focus on the dog and the cat in the MS COCO2017 dataset. The dog and the cat belong to the same major class and have high similarity to each other, which is like the mud and ore. The training set of the MS COCO contains 4385 images with 5508 label boxes for the dog and 4114 images with 4768 label boxes for the cat. Each image is 608×608 pixels.

The Ore Dataset is used to fine-tune the YOLO-v3model. It is collected from a real mine and labeled by the experienced workers. Since the actual production is more focused on a larger object, there is no label for objects with diameters less than 50 mm. It contains 5683 images, and each image is 720×640 pixels. The Ore Dataset is also organized with the format of the MS COCO. Different from the MS COCO, each image in the Ore Dataset contains ore object, but partly contains mud object. The detail is shown in Table 1.

As shown in Figure 5, the scene of the Ore Dataset is more complex than the MS COCO. One image often contains multiple pieces of mud and ores with large-scale change. The inhomogeneous slurry makes the background more complex. Nonuniform illumination and the occlusion between ore and mud further complicate the scene. So, detecting mud from the ores is more challenging than dog or cat detection.

4.2. Experiment Design. In order to verify the effectiveness of the proposed method, this paper designs comparison experiments with YOLO-v3 method, YOLO-v3 (RS) method, and YOLO-v3 (LC).

(1) *YOLO-v3.* We pretrain a YOLO-v3 model with MS COCO Dataset without cat and dog samples and fine-tune the model with the dog and cat dataset and the Ore Dataset, respectively. Different from the YOLO-AL method that iteratively increases labeled samples, it uses all samples at one time for training a fine YOLO-v3 model. Then, we observe the detection performance and compare with the YOLO-AL method.

To train the YOLO-v3 model, we take the default hyperparameters. The ratios of training and testing samples are 0.7 and 0.3, respectively. The experimental results are shown in Table 2 and are marked with a red* in Figure 6. It is worth noting that all samples were used for model training and testing at one time, which is significantly different from the increasing training samples in the YOLO-AL method.

(2) *YOLO-AL (RS).* The YOLO-AL (RS) method with the RS strategy randomly selects samples for training. It treats the training sample indiscriminately, which is the same as that in experiment 1. So, they are essentially the same with each other. The only difference is that the RS increases the labeled samples gradually, while the YOLO-v3 uses all samples at one time. However, the RS can still find out exactly how many training samples are enough for training a model.

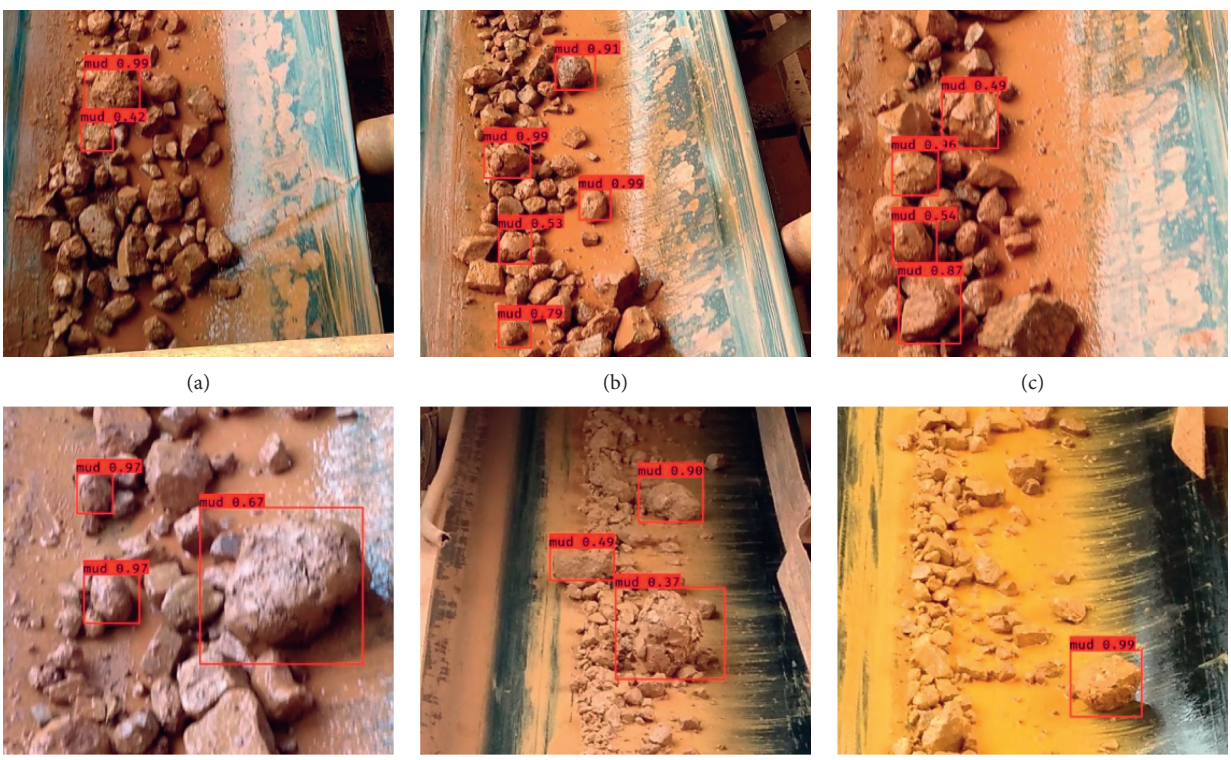
On the cat and dog dataset and the Ore Dataset, the YOLO-AL (RS) was trained with the sample increment $h = 50$. For simplicity, the expert verification is

Inputs:
I_L (labeled training image sample set)
I_U (unlabeled image set)
$M_{\text{yolo}-v3}$ (the pretrained YOLO-v3 model)
h (the sample increment)
Outputs:
$M_{\text{yolo}-v3}$ (fine-tuned by AL method)
(1) Train the YOLO-v3 model $M_{\text{yolo}-v3}$ with the labeled training image sample set I_L and update $M_{\text{yolo}-v3}$ and YOLO-v3 detector.
(2) Detect objects for each image in I_U and calculate confidence for each object.
(3) Sort the mud objects in ascending order according to $p_{\text{uncertain}}$ and take the first n mud objects contained in h images which is composed of sample set $I_{\text{uncertain}}$.
(4) For each image in $I_{\text{uncertain}}$, the expert checks the object label or box bound and makes appropriate modification. The images with verified labels form sample set I_v.
(5) Add verified samples I_v to the current training set I_L and remove them from I_U.
(6) Continue step 1 to step 5 till the set I_U is null or reaches the specified number of iterations.

Algorithm 1: The YOLO-v3-based AL framework.

Table 1: The samples of the datasets.

	MS COCO dog and cat		Ore Dataset	
	Dog	Cat	Mud	Ore
Images	4385	4114	4375	5683
Label boxes	5508	4768	14345	45742



(a) (b) (c)

(d) (e) (f)

Figure 5: Continued.

(g) (h) (i)

Figure 5: Results of the mud detection.

Table 2: The detection accuracy (AP) of the methods.

	Dog and cat/AP (%)		Ore/AP (%)	
	Dog	Cat	Mud	Ore
YOLO-v3	72.5	72.5	73.8	66.4
YOLO-AL (RS)	**72.9**	72.4	73.7	66.5
YOLO-AL (LC)	72.8	**72.6**	**75.1**	**68.3**

done instead by sample querying during the experiment. Namely, the labels of the unlabeled samples are hidden first, and when the selected samples are sent to the expert for verification, the corresponding labels are queried from the label set.

(3) *YOLO-AL (LC)*. The YOLO-AL (LC) method with the LC strategy selects samples. The other setting is the same as the YOLO-AL (RS).

4.3. Results and Analysis. To assess the effectiveness of the proposed YOLO-AL model, the average precision (AP) and mean average precision (mAP) are adopted. The AP measures the quality of bounding box prediction in the test set. If the IoU of a predicted box with the ground truth is larger than 0.5, the prediction is considered as true positive [26].

In order to avoid the randomness of detection performance, we performed five experiments for each method. Then, we calculated the mean and standard deviation of the mAP to form Figures 6 and 7, where the colored background areas represent the standard deviation floating range.

In Figures 6 and 7, the x coordinate is sample number with the increment $h = 50$, while the y coordinate is the mAP. As shown in Figure 6, the converged mAPs of the three methods have no obvious difference on the dog and cat dataset, which can also be seen from Table 2. However, the required training samples in the proposed methods are much less than those in the YOLO, as shown in Table 3. The required samples of the three methods are 2350, 3100, and 4400, respectively. The YOLO-AL (LC) is about 53.4% of the YOLO and 70.5% of the YOLO (RS).

On the Ore Dataset, the YOLO-AL (RS) is not obviously different from the YOLO-AL (LC). However, the accuracy of the YOLO-AL (LC) is about 1.5% higher than the other. The result is amazing due to the complex scene and the low discrimination between ore and mud. The required samples of the three methods are 1950, 2650, and 4400, respectively. The YOLO-AL (LC) is about 44.3% of the YOLO and 73.6% of the YOLO (RS).

The following conclusions can be drawn. (1) The detection of accuracy of the proposed method is no less than that of the YOLO-v3. (2) The required training samples of the proposed method are obviously less than those of the YOLO-v3. (3) The proposed method can be easily transferred from one mine to another. On the one hand, the proposed method uses the most valuable samples to fine-tune a model which needs less labeled samples. On the other hand, the difference between mines is less than that between Ore Dataset and COCO dataset, so it needs fewer labeled samples to transfer the model from one mine to another.

The main reason maybe lies in the training process. The AL (LC) can pick the most uncertain samples, which are most valuable for model training. In other words, the sample that cannot be accurately "understood" by the current model can provide meaningful information for improving model accuracy. The samples that cannot be accurately "understood" by the current model provide only a little meaningful information and can even be ignored.

Another reason may be that the AL with LC strategy can prevent samples from being overconcentrated in a certain area of the feature space, which may lead to biased estimates.

With the model trained by the proposed method, the Ore Dataset was tested. Partial results of the mud detection are shown in Figure 5. For the sake of clarity, the bounding boxes of the ore object are hidden here. Although the scene is complex and the mud and ore are only slightly different in color, texture, and shape, the proposed method can effectively distinguish ore and mud.

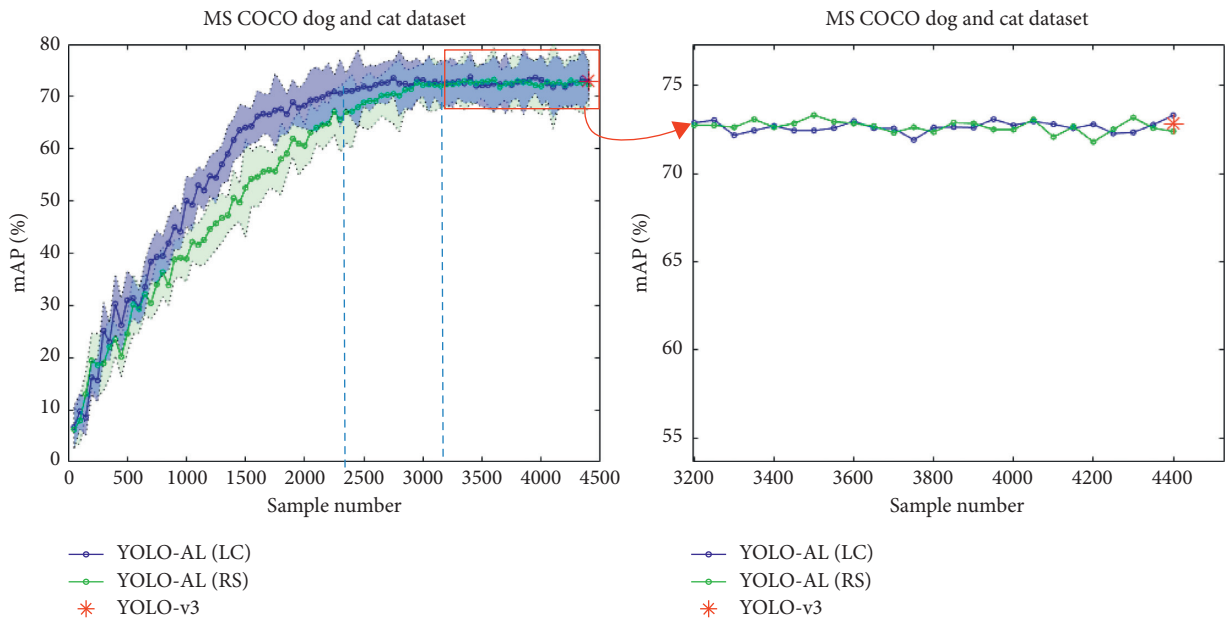


Figure 6: The mAP-sample number curves of the MS COCO dog and cat dataset.

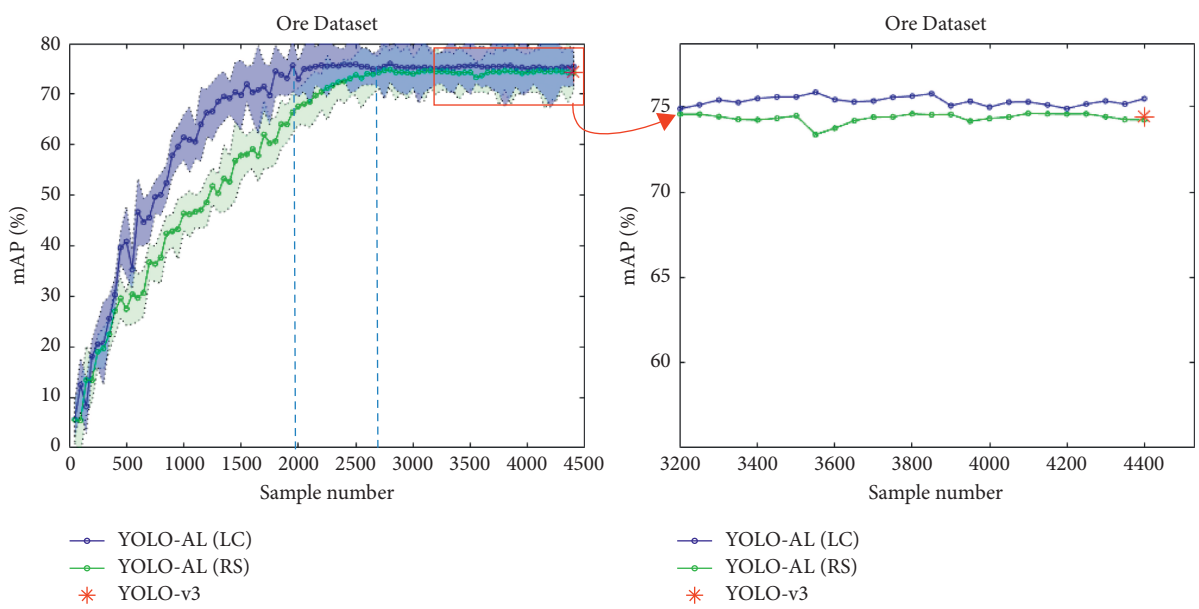


Figure 7: The mAP-sample number curves of the Ore Dataset.

Table 3: The required samples when the model converges.

	Dog and cat mAP = 73.1%	Ore mAP = 71.8%
YOLO-v3	4400	4400
YOLO-AL (RS)	3100	2650
YOLO-AL (LC)	2350	1950

The detection speed of this method is close to that of YOLO-v3. Our personal computer is 64 bit Windows 8.1 system, with Intel Core i5 CPU, 2.60 Hz, and 8 GB RAM. On the COCO dataset, the detection speed is about 30 fps, while on the Ore Dataset, the detection speed is about 28 fps. It is because the image of the Ore Dataset is a little larger than that from the COCO dataset.

5. Conclusions and Future Work

Automatically detecting mud in bauxite ores is valuable and challenging. This paper proposed a novel weakly supervised method which combines the deep active learning and the YOLO-v3 model. To select the most valuable samples, it adopts the Less Confident (LC) strategy according to the confidences of objects predicted by the YOLO-v3 detector. Then, it fine-tunes the model in the AL process with the

valuable samples every time. Experimental results show that the proposed method can effectively detect mud in ores. More importantly, the proposed method needs much fewer labeled samples than YOLO-v3 without decreasing the detection accuracy, which extensively reduces human labor for annotating training data. Also, the proposed method can be easily transferred from one mine to another, which is important for the practical application of mud detection.

In future work, we will study more appropriate sample selection strategies to further reduce the labeling cost. In addition, ores containing more types of impurities will be considered.

Data Availability

The ore images and labeled data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 12 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Scientific Research Fund of Hunan Provincial Education Department (nos. 18A376 and XJK17BXX010) and National Natural Science Foundation of China (no. 11701172).

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2015.

[2] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, December 2015.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 91–99, Montreal, Canada, December 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2014.

[5] W. Liu, D. Anguelov, and D. Erhan, *SSD: Single Shot Multibox Detector. European Conference on Computer Vision*, Springer, Cham, Switzerland, 2016.

[6] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: object detection via region-based fully convolutional networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 379–387, Vancouver, Canada, December 2016.

[7] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, Salt Lake City, UT, USA, June 2018.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[9] Redmon J., Farhadi A., YOLO9000: Better, Faster, Stronger, arXiv preprint, 2017.

[10] Redmon J., Farhadi A., Yolov3: An Incremental Improvement, arXiv preprint arXiv:1804.02767, 2018.

[11] M. M. Crawford, D. Tuia, and H. L. Yang, "Active learning: any value for classification of remotely sensed data?" *Proceedings of the IEEE*, vol. 101, no. 3, pp. 593–608, 2013.

[12] B. Settles, "Active learning literature survey," Technical Report 1648, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, USA, 2010.

[13] S. J. Huang, R. Jin, and Z. H. Zhou, "Active learning by querying informative and representative examples," in *Proceedings of the International Conference on Neural Information Processing Systems*, Vancouver, Canada, December 2010.

[14] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active learning with gaussian processes for object categorization," in *Proceedings of the 2007 IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2015.

[15] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive annotation: a deep active learning framework for biomedical image segmentation," in *Medical Image Computing and Computer Assisted Intervention—MICCAI 2017*, pp. 399–407, Springer, Berlin, Germany, 2017.

[16] S. Sun, Z. Ping, H. Xiao, and R. Wang, "A MRF model-based active learning framework for the spectral-spatial classification of hyperspectral imagery," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 6, pp. 1074–1088, 2015.

[17] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class Active learning by uncertainty sampling with diversity maximization," *International Journal of Computer Vision*, vol. 113, no. 2, pp. 113–127, 2015.

[18] S. Dutt Jain and K. Grauman, "Active image segmentation propagation," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2864–2873, Las Vegas, NV, USA, June 2016.

[19] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3325–3337, 2015.

[20] P. Zhou, G. Cheng, Z. Liu, S. Bu, and X. Hu, "Weakly supervised target detection in remote sensing images based on transferred deep features and negative bootstrapping," *Multidimensional Systems and Signal Processing*, vol. 27, no. 4, pp. 925–944, 2016.

[21] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 265–278, 2018.

[22] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7405–7415, 2016.

[23] J. Yosinski, J. Clune, Y. Bengio et al., "How transferable are features in deep neural networks?" in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3320–3328, Montreal, Canada, 2014.

[24] B. Settles, "Active learning literature survey. computer sciences," Technical Report 1648, University of Wisconsin–Madison, Madison, WI, USA, 2009.

[25] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: common objects in context," in *European Conference on Computer Vision*, pp. 740–755, Springer, Berlin, Germany, 2014.

[26] T. Deselaers, B. Alexe, and V. Ferrari, "Weakly supervised localization and learning with generic knowledge," *International Journal of Computer Vision*, vol. 100, no. 3, pp. 275–293, 2012.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1204260, 21 pages
https://doi.org/10.1155/2020/1204260

Research Article

Optimal Torque Split Strategy of Dual-Motor Electric Vehicle Using Adaptive Nonlinear Particle Swarm Optimization

Qingxing Zheng,[1,2,3,4] **Shaopeng Tian**,[1,2,3,4] **and Qian Zhang**[1,2,3,4]

[1]*School of Automotive Engineering, Wuhan University of Technology, Wuhan 430070, China*
[2]*Hubei Key Laboratory of Advanced Technology for Automotive Components, Wuhan University of Technology, Wuhan 430070, China*
[3]*Hubei Collaborative Innovation Center for Automotive Components Technology, Wuhan 430070, China*
[4]*Hubei Research Center for New Energy & Intelligent Connected Vehicle, Wuhan University of Technology, Wuhan 430070, China*

Correspondence should be addressed to Shaopeng Tian; tianshp@whut.edu.cn

Received 16 January 2020; Revised 30 March 2020; Accepted 7 April 2020; Published 21 May 2020

Guest Editor: Chi-Hua Chen

In order to exploit the potential of energy saving of dual-motor powertrain over single-motor powertrain, this paper proposes a time-efficient optimal torque split strategy for a front-and-rear-axle dual-motor electric powertrain. Firstly, a physical model of electric vehicle powertrain is established in Matlab/Simulink platform and further validated by real-vehicle experiments. Subsequently, a three-layer energy management strategy composed of demanded torque calculation layer, mode decision layer, and torque split layer is devised to enhance the total operating efficiency of two motors. Specifically, the optimal torque split strategy using adaptive nonlinear particle swarm optimization (ANLPSO) is embedded in the torque split layer. Finally, two conventional strategies (even distributed strategy and rule-based strategy) for dual-motor powertrain are considered for comparison to verify the efficacy of the proposed strategy. Tremendous results demonstrate that the dual-motor powertrain with this proposed optimal torque split strategy develops energy saving by 11.88% and 12.18% against single-motor powertrain in the NEDC and WLTP. Compared to two conventional torque split strategies, it is able to reduce the total motor loss by 12.17% and 8.1% in NEDC and 11.91% and 8.07% in WLTP, respectively, which indicates the prominent optimization performance and a great potential in realistic applications.

1. Introduction

To relieve increasing burden caused by environmental pollution and energy shortage, electric vehicles (EVs) emerged as a promising approach all over the world [1]. While the energy efficiency of internal combustion engine (ICE) vehicles is merely from 15% to 20%, typical EVs utilize 50–75% of the energy effectively to propel the vehicle [2, 3]. There are three main categories of EVs, including hybrid electric vehicles (HEVs), fuel cell electric vehicles (FCHEVs), and battery electric vehicles (BEVs) [4], among which BEVs have the best "tank-to-wheel" efficiency [5]. Furthermore, the salient upsides of no CO_2 emissions and other pollutants lead BEVs to be one of the most promising alternatives for green transportation and sustainable future mobility [6]. However, BEVs are still facing driving range concern, mainly owing to the long charging time and unsatisfactory energy density of power battery when compared to gasoline and diesel [7].

1.1. Literature Review. To improve the driving range of the conventional single-motor-driven BEVs, from the perspective of powertrain topology and control, various powertrain structures and relevant control strategies have been proposed and investigated. These advanced configurations offer additional degrees of freedom to achieve more efficient torque distribution and independent control of each driving

shaft/wheel. The literature review mainly focuses on two following aspects: powertrain topology and energy management strategy.

1.1.1. Powertrain Topology. According to the number of motors used in the vehicle and their configuration, existing powertrain topology of BEVs can be classified into two types including the centralized single-motor-driven powertrain and the distributed multimotor-driven powertrain.

The centralized single-motor one-speed-driven powertrain is the most common structure in modern BEVs. As the motor substitutes the ICE to propel the vehicle through a one-speed reducer, it can be easily installed on the conventional chassis without excessive modification. Compared to one-motor multispeed powertrain, this kind of powertrain is known as the simplest structure, whereas it would increase the working burden of both motor and controller. Moreover, it has relatively low energy efficiency [8]. To address such drawbacks, some researchers proposed the single-motor with multispeed gearbox powertrain. The results indicated that multispeed gearbox could downsize the motor and improve the overall energy efficiency by about 5–12% through the improvement of motor operating points [9, 10]. However, this powertrain is rarely used in practice because the additional multispeed gearbox might cause the increase in cost and control difficulty of shift smoothness (especially with the increase in gears) [11].

Recently, some researchers have focused on distributed multimotor-driven powertrain. Existing multimotor-driven powertrain is mainly classified into three categories [12]: dual-motor powertrain, triple-motor powertrain (shown in Figure 1(c)) [13], and four-motor powertrain (shown in Figure 1(d)) [14–16]. Specially, dual-motor powertrain has been actively investigated as a promising alternative to single-motor powertrain. Hu et al. studied a novel dual-motor powertrain in which the speed of two motors is coupled through planetary gear and the torque is coupled through shaft-fixed gear. The simulation shows that this powertrain could improve energy efficiency effectively [17]. However, the speed and torque coupling devices increase structure complexity, manufacturing cost, and control difficulty. Gao et al. researched a practical dual-motor coaxial powertrain for battery electric buses with dynamic programming (DP) method, which could eliminate power interruption and realize a great reduction of energy consumption [18]. Actually, it is one kind of planet gear-based coupling system, and additional coupling devices, such as clutches and synchronizers, are still needed, which burden the mechanism design and control. In previous study [19], there is another type of powertrain with two wheel-hub motors directly driving the left/right wheel through a one-speed reducer, shown in Figure 1(b). But this powertrain is rarely used in practice. Ruan and Song devised a novel dual-motor two-speed powertrain as one of the parallel direct drive layouts for BEVs. This structure could achieve a great efficiency improvement and smooth shifting process without the increase in mechanical complexity [20]. Kang et al. proposed an electric powertrain with the front and rear shaft driven by two motors, respectively (shown in Figure 1(a)). It exhibits great enhancement of drivability and potential of energy saving [21]. Due to the low manufacturing cost, simple architecture, energy efficiency improvement potential, and great dynamics performance, the front-and-rear-axle dual-motor powertrain is investigated in this paper.

1.1.2. Energy Management Strategy. In dual-motor powertrain, energy management strategy plays an important role in power split among two motors and energy saving. Hu et al. proposed the multimode control strategy and power split strategy for a dual-motor powertrain with a planetary gear. The results show great improvement in economic performance and driving range [18]. Urbina Coronado et al. designed a two-parameter rule-based control strategy for a dual-motor powertrain. The results indicate that a combination of high-speed transition threshold and high torque transition threshold could achieve the best results in energy saving and drivability [22]. In previous study [23], theoretical solutions of power loss optimization were conducted for a front-rear-induction-motor EVs. It is confirmed that single-motor mode takes priority over dual-motor mode under required low torque condition, while even torque distribution is the optimal control in dual-motor mode under required middle or high torque condition. Yuan and Wang investigated the optimal torque distribution strategy for an EV with two identical PMSMs to improve motor efficiency in the region of low torque and high speed. This study reveals that the motor efficiency could be significantly improved in low torque region when one motor is operating and the other motor is fully turned off through a clutch. And in the region of middle or high torque, the required torque should be equally distributed to minimize power loss when only the longitudinal dynamics is considered [24]. Wang et al. suggested a mode decision system based on fuzzy logic for a dual-motor powertrain. The frequent mode switching is reduced by the defined offset and compensation for mode switching line [25]. Zhao et al. developed a novel rule-based strategy extracted from DP optimization results. The simulation results show that the improved strategy could reach 95% energy-saving effect of that under DP [26].

1.2. Motivation and Innovation. This study was carried out in order to prove the potential of energy efficiency improvement of a front-and-rear-axle dual-motor powertrain over single-motor powertrain. In aforementioned literature, tremendous research studies were conducted on control strategy for front-and-rear-axle-driven powertrain. It is concluded that the switching threshold between single-motor mode and dual-motor mode is related to motor speed. In practice, the motor speed varies with unexpected road conditions, which requires the switching threshold and torque split strategy to be optimized in real time for better control. However, a vast majority of existing torque split strategies are designed based on theoretical analysis or empirical rules, using a fixed torque switching threshold for control simplicity but with the sacrifice of optimality.

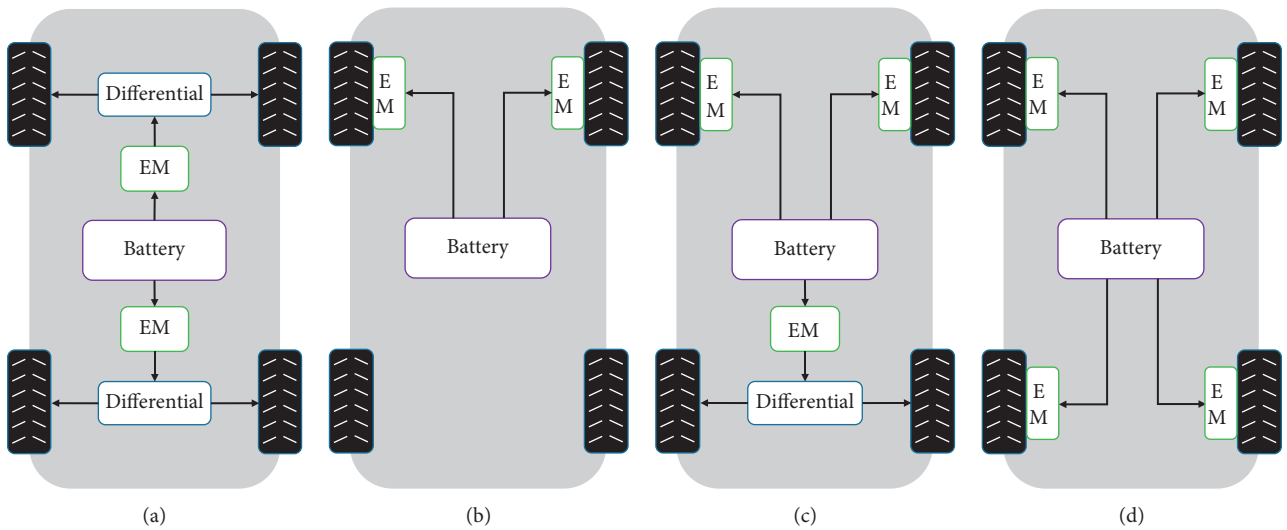


Figure 1: BEV powertrains.

To address this research gap, this paper developed a time-efficient optimal three-layer energy management strategy based on adaptive nonlinear particle swarm optimization (ANLPSO). The main contributions incorporate three aspects. Firstly, the optimal torque split strategy was established to search the best torque split ratio of two motors instantaneously and thus achieve the optimal energy efficiency. Secondly, the algorithm architecture and main influencing factors of PSO were improved to construct the proposed ANLPSO, which could perform a tradeoff between optimality and real-time applicability. Eventually, a comprehensive comparison was conducted between two common torque split strategies and the optimal one.

The organization of this paper is as follows: the powertrain architecture is described and modeled in Section 2. The three-layer energy management strategy and the real-time optimal torque distribution strategy are designed in Section 3. In Section 4, the physical model is validated by real-vehicle experiment, followed by the comparison and analysis of the dual-motor powertrain simulation results between even distribution strategy, the rule-based strategy in literature, and the proposed real-time optimal control strategy. Finally, conclusions are summarized in Section 5.

2. Powertrain Architecture and Model

2.1. Electric Powertrain Configuration. Figure 1(a) illustrates the configuration of the front-and-rear-axle dual-motor electric powertrain. In this architecture, two identical PMSMs are coupled to the front axle or the rear axle through a one-speed gear set and a differential mechanism, respectively. The target dual-motor BEV in this paper is a commercial vehicle prototype (shown in Figure 2), with all relevant parameters detailed in Tables 1–3.

2.2. Vehicle Longitudinal Dynamics Model. This research focuses on the energy efficiency of the dual-motor powertrain. Thus, only the longitudinal dynamics vehicle model is studied in this paper because the vehicle vertical vibration and vehicle yaw have little influences on energy efficiency. Generally, the overall resistance force over wheels consists of air resistance, rolling resistance, and climbing resistance. However, in this study, the climbing resistance is neglected since the driving cycles are horizonal. Hence, the final resistance force over wheels is calculated by equations shown below:

$$F_r = \frac{C_d A v(k)^2}{21.15} + mgC_r + \frac{J_t a(k)}{r^2}, \tag{1}$$

$$J_t = J_v + J_w + (J_{\mathrm{EM1}} + J_{\mathrm{EM2}}) i_t^2, \tag{2}$$

$$m\frac{\mathrm{d}v(k)}{\mathrm{d}t} = \frac{(T_{\mathrm{EM1}} + T_{\mathrm{EM2}}) i_t \eta_t}{r} - F_r, \tag{3}$$

where F_r is the overall resistance force over wheels; C_d is the air drag coefficient; A is the frontal area of vehicle; v is the velocity of vehicle; k is the kth time step; m is the mass of vehicle; g is the gravitational acceleration; C_r is the rolling resistance coefficient; a is the longitudinal acceleration of vehicle; J_t is the total rotary inertia at the wheels; J_v is the rotary inertia of vehicle chassis; J_w is the total rotary inertia of four wheels; J_{EM1} and J_{EM2} are the rotary inertia of EM1 and EM2, respectively; T_{EM1} and T_{EM2} are the output torque of EM1 and EM2, respectively; i_t is the transmission ratio; η_t is the transmission efficiency; and r is the rolling radius of wheel.

2.3. Driver Model. The driver model is designed based on feedforward-feedback PI controller. Based on the bias between the desired velocity (given by driving cycle) and the actual velocity, acceleration or braking command will be given by the driver model. To improve the control precision and reduce the fluctuation of driver commands, a feedforward control module based on desired velocity, a feedback control module based on driver commands, and an

Figure 2: Target dual-motor BEV.

Table 1: Vehicle parameters of target vehicle.

Vehicle characteristics	
Vehicle mass (half load) (m)	2000 kg
Dynamic rolling radius (r)	0.273 m
Frontal area (A)	5 m^2
Transmission efficiency (η_t)	0.9504
Air drag coefficient (C_d)	0.4
Rolling resistance coefficient (C_r)	0.01
Rotary mass coefficient (J_t)	1.12
Gravitational acceleration (g)	9.8 m/s^2
Final drive ratio (front/rear axle)	6.732
Top speed	85 km/h
Climbing capacity	20% at 25 km/h
Acceleration time (0–50 km/h)	7 s

Table 2: Parameters of driving motor.

Name	Value
Type	PMSM (ETG029)
Rated/peak power	15/44.6 kW
Rated/peak torque	31/117 Nm
Maximum rotation speed	7500 rpm
Nominal voltage	115 V
Peak efficiency	96%

Table 3: Parameters of power battery.

Name	Value
Type	$LiFePO_4$
Capacity (C)	375 Ah
Nominal voltage	115.2 V
Operating temperature range	−20–55°C
Mass	320 kg

additional high-pass filter were added. The driver command can be calculated by the following equation:

$$\begin{aligned} \mathrm{Acc}_{\mathrm{pdl}}(k) &= K_{FF}V_d(k) + K_p\left(V_d(k) - V_a(k)\right) \\ &\quad + \int_0^k \left(K_i\left(V_d(k) - V_a(k)\right) + K_{FB}D_c\right)\mathrm{d}k, \\ &\quad V_d(k) - V_a(k) \geq 0, \end{aligned} \tag{4}$$

$$\begin{aligned} \mathrm{Brk}_{\mathrm{prs}}(k) &= G_{\mathrm{brk}} * \left(K_{FF}V_d(k) + K_p\left(V_d(k) - V_a(k)\right)\right. \\ &\quad \left. + \int_0^k \left(K_i\left(V_d(k) - V_a(k)\right) + K_{FB}D_c\right)\mathrm{d}k\right), \\ &\quad V_d(k) - V_a(k) < 0, \end{aligned} \tag{5}$$

where $\mathrm{Acc}_{\mathrm{pdl}}$ is the acceleration pedal command; G_{brk}, K_{FF}, K_p, K_i, and K_{FB} are constant gain of the demanded brake pressure, feedforward module, proportional gain, integral gain, and feedback module, respectively; $V_d(k)$ is the desired velocity; $V_a(k)$ is the actual velocity; D_c is the driver command given by driver model; and $\mathrm{Brk}_{\mathrm{prs}}$ is the demanded brake pressure.

2.4. Driving Motor Model. The driving motors in this powertrain consist of two identical PMSMs, which can work as the driving motor to propel the vehicle or as a generator to recover braking energy. The data of motor external characteristics and efficiency characteristics are based on experimental data, which is shown in Figure 3. The output torque of motor is simplified as a function of motor speed and motor torque command, as shown in equation (6). The efficiency characteristics are related to motor speed and motor output torque, as shown in equation (7):

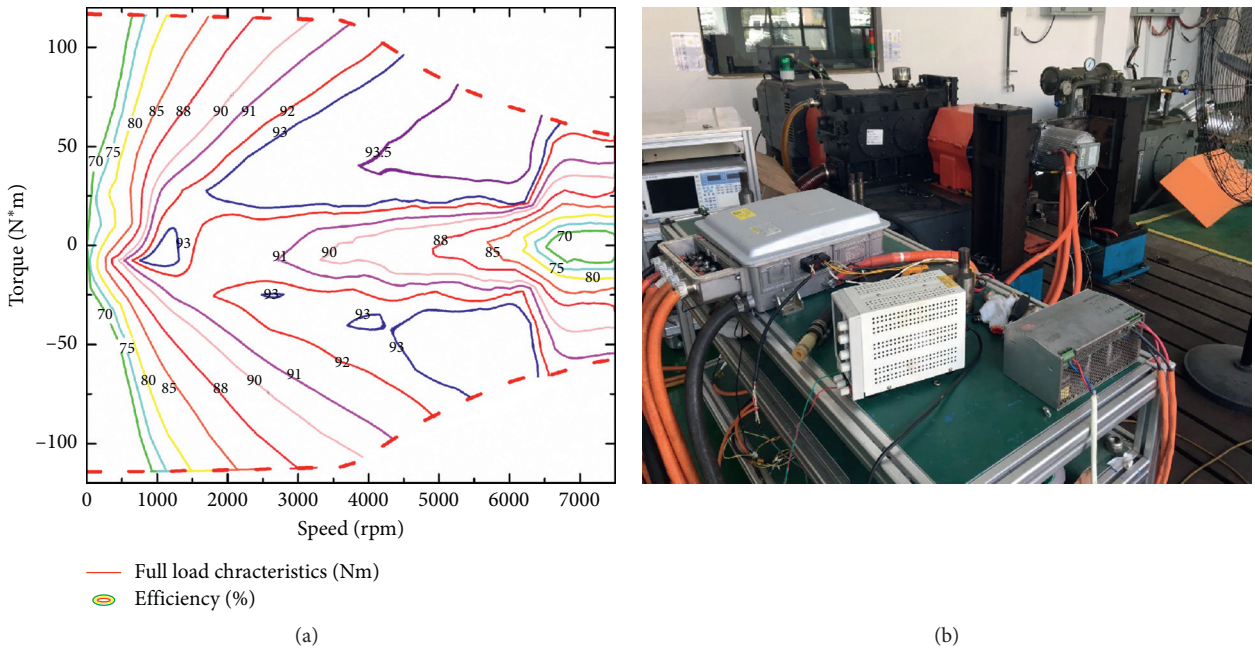


Figure 3: (a) External characteristics and efficiency characteristics; (b) motor test bench.

$$T_{\mathrm{EM}i}(k)=\begin{cases} l_{\mathrm{EM}i}(k)\times T_{\mathrm{chgmaxEM}i}(n_{\mathrm{EM}i}(k)), & l_{\mathrm{EM}i}(k)<0; i=1,2, \\ l_{\mathrm{EM}i}(k)\times T_{\mathrm{dismaxEM}i}(n_{\mathrm{EM}i}(k)), & l_{\mathrm{EM}i}(k)\geq 0; i=1,2, \end{cases} \tag{6}$$

$$\eta_{\mathrm{EM}i}(k)=f\left(n_{\mathrm{EM}i}(k),T_{\mathrm{EM}i}(k)\right), \quad i=1,2, \tag{7}$$

$$P_{\mathrm{EM}i}(k)=\begin{cases} \dfrac{n_{\mathrm{EM}i}(k)T_{\mathrm{EM}i}(k)}{9550\eta_{\mathrm{EM}i}(k)}, & T_{\mathrm{EM}i}(k)\geq 0; i=1,2, \\ \dfrac{n_{\mathrm{EM}i}(k)T_{\mathrm{EM}i}(k)\eta_{\mathrm{EM}i}(k)}{9550}, & T_{\mathrm{EM}i}(k)<0; i=1,2, \end{cases} \tag{8}$$

where n_{EM} is the motor speed, T_{EM} is the output torque of motor, l_{EM} is the motor torque command, $T_{\mathrm{chgmaxEM}}(n_{\mathrm{EM}})$ is the maximum charge torque of motor at n_{EM}, $T_{\mathrm{dismaxEM}}(n_{\mathrm{EM}})$ is the maximum discharge torque of motor at n_{EM}, η_{EM} is the efficiency of motor, and P_{EM} is the electric power of motor.

2.5. Battery Model. The electrochemistry reaction inside the battery is quite complicated, which is not the scope of this study. Hence, the open equivalent circuit model [27] is adopted, which reflects the relationship between open circuit voltage, inner resistance, and the state of charge (SOC). In this case, when the battery output power is given, the current of battery and SOC can be calculated as follows:

$$P_b(k)=\sum_{i=1}^{2}P_{\mathrm{EM}i}(k)+I(k)^2R(\mathrm{SOC}(k)), \tag{9}$$

$$I(k)=\frac{P_b(k)}{U_{\mathrm{OC}}(\mathrm{SOC}(k))}, \tag{10}$$

$$\mathrm{SOC}(k)=\mathrm{SOC}_{\mathrm{ini}}-\frac{\int_0^k I_b(k)\mathrm{d}k}{3600C}, \tag{11}$$

where P_b is the total battery power, I is the battery current, $R(\mathrm{SOC}(k))$ is the internal resistance at SOC(k), $U_{\mathrm{OC}}(\mathrm{SOC}(k))$ is the open circuit voltage at SOC(k), $\mathrm{SOC}_{\mathrm{ini}}$ is the initial value of SOC, and C is the battery capacity.

2.6. Driving Cycle. The driving cycle is a standard test procedure in which the desired velocity is fixed with respect to time. The driver is supposed to propel the vehicle to reach the desired velocity within a certain error tolerance range to ensure accuracy of the test or simulation. Numerous driving cycles are proposed to evaluate energy consumption, fuel consumption, and emissions for sorts of vehicles in different countries, such as New European Driving Cycle (NEDC), Urban Dynamometer Driving Schedule (UDDS), Japan Cycle 08 (JC08), World Light Vehicle Test Procedure (WLTP). In this research, NEDC and WLTP are selected as the standard test cycle for the target vehicle. As mentioned above, the target vehicle is a commercial sample vehicle with a maximum velocity of 85 km/h. Therefore, the portions with the velocity exceeding 85 km/h in NEDC and WLTP are limited to 85 km/h. Figure 4 shows the driving cycles with velocity limitation.

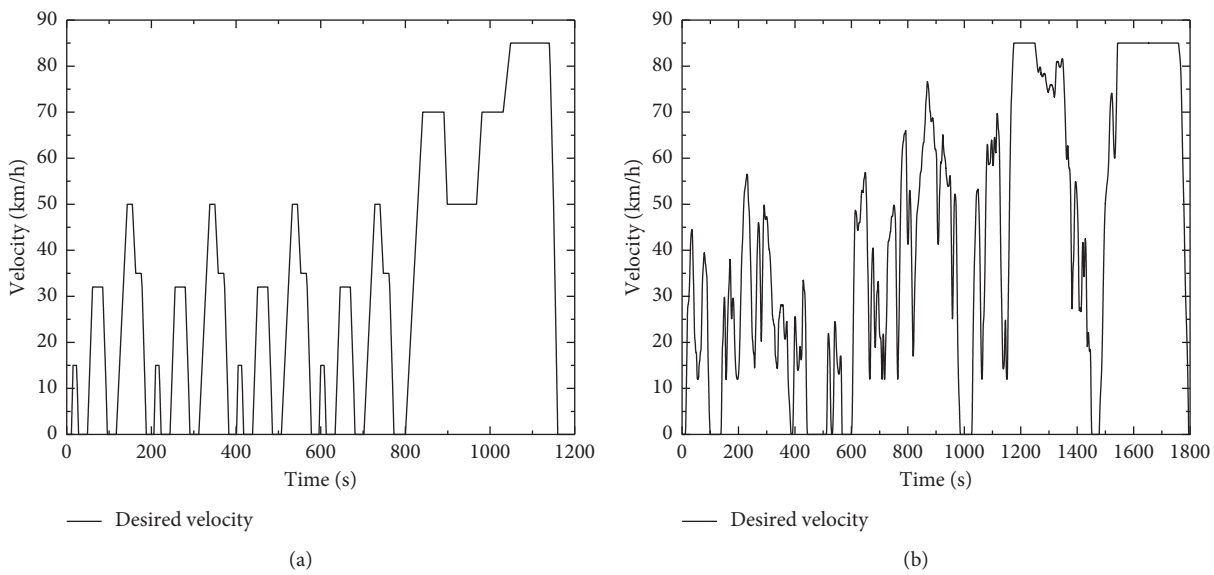


Figure 4: (a) NEDC with velocity limitation; (b) WLTP with velocity limitation.

3. Optimal Energy Management Strategy

To optimize energy efficiency and dynamics performance, energy management strategy plays a paramount role in power split for dual-motor powertrain. In this research, the optimal energy management strategy consists of three layers: demanded torque calculation (upper layer), vehicle operating mode decision (decision layer), and real-time optimal torque split (lower layer), with the schematic diagram shown in Figure 5.

3.1. Demanded Torque Calculation. In the upper layer, the demanded torque can be calculated based on the driver's command, state of two motors, and sensor signals of hydraulic brake. In detail, the demanded driving torque is determined by the acceleration pedal signal and the available maximum output torque of two motors, while the demanded brake pressure is decided by driver's brake command and constant gain of the demanded brake pressure (as mentioned in Section 2.3).

3.2. Vehicle Operating Mode Decision. The decision layer is designed to define the vehicle operating mode based on driving torque command, brake pressure, SOC of battery, and current velocity. The simple and robust "if-else" control strategy is adopted to yield the most appropriate operating mode for this powertrain through state machine. The mode switching logic is illustrated in Figure 6.

Electric mode: if the demanded driving torque is greater than zero and the battery SOC is greater than discharge lower limit (SOC_l), the vehicle is in electric mode. And the magnitude of the demanded driving torque determines whether the vehicle is driven by a single motor or two motors, which is further determined by the torque split strategy.

Regenerative brake mode: if the battery SOC is below charge upper limit (SOC_h) and the demanded brake torque is nonzero and not greater than the available maximum torque of two motors and the current velocity is greater than a fixed value, the vehicle will operate in the regenerative brake mode. In order to adapt to different braking requirements, the regenerative brake mode is further divided into three sub-operating modes including energy recovery priority mode to maintain high regenerative braking efficiency, braking performance priority mode to obtain maximum braking force, and safety priority mode for the sake of safety in wet/snow road surface [28]. In this study, the energy recovery priority mode is the main working mode for regenerative braking.

Hydraulic brake mode: considering vehicle safety, whenever the demanded brake torque exceeds the threshold defined by the ECE safety regulation or the current velocity is smaller than a specified value, the hydraulic brake mode is switched on.

3.3. Real-Time Optimal Torque Split Strategy. The lower layer is torque split strategy, composed of the optimal driving torque split strategy and braking torque split strategy. However, in this paper, the proposed real-time optimal driving torque split strategy plays the major role and would be detailed further. At the same time, to explore its potential, the even distribution strategy and the control strategy proposed by some researchers would be included for comparison in terms of energy efficiency.

3.3.1. Baseline Driving Torque Split Strategy. For dual-motor powertrain, the simplest and most common torque split strategy is even distribution strategy, in which the demanded driving torque is always distributed equally to two motors to

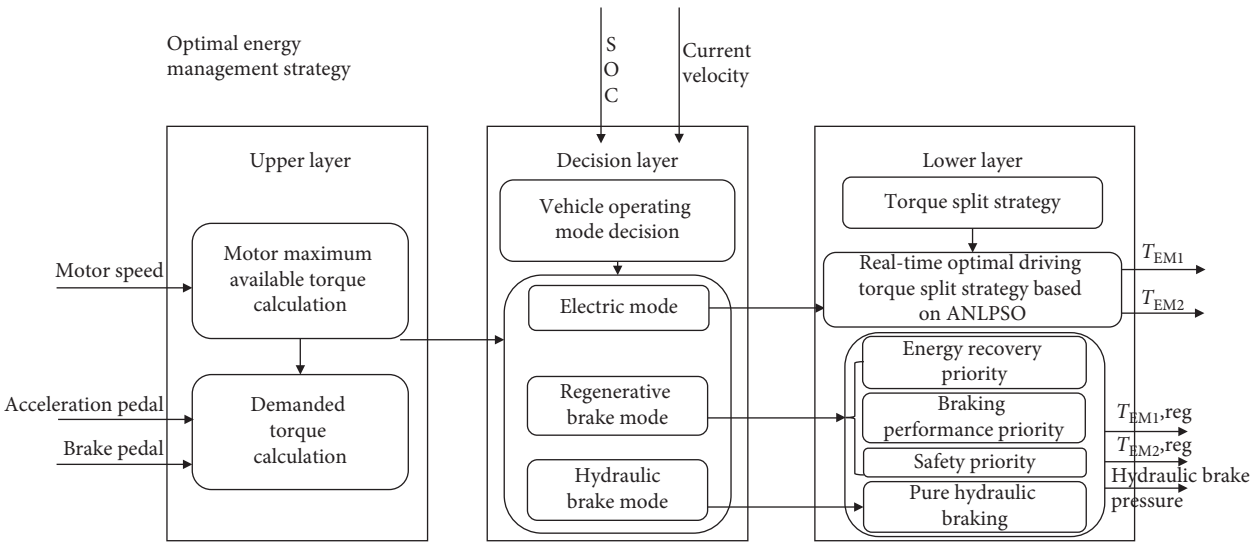


Figure 5: Schematic of optimal energy management strategy.

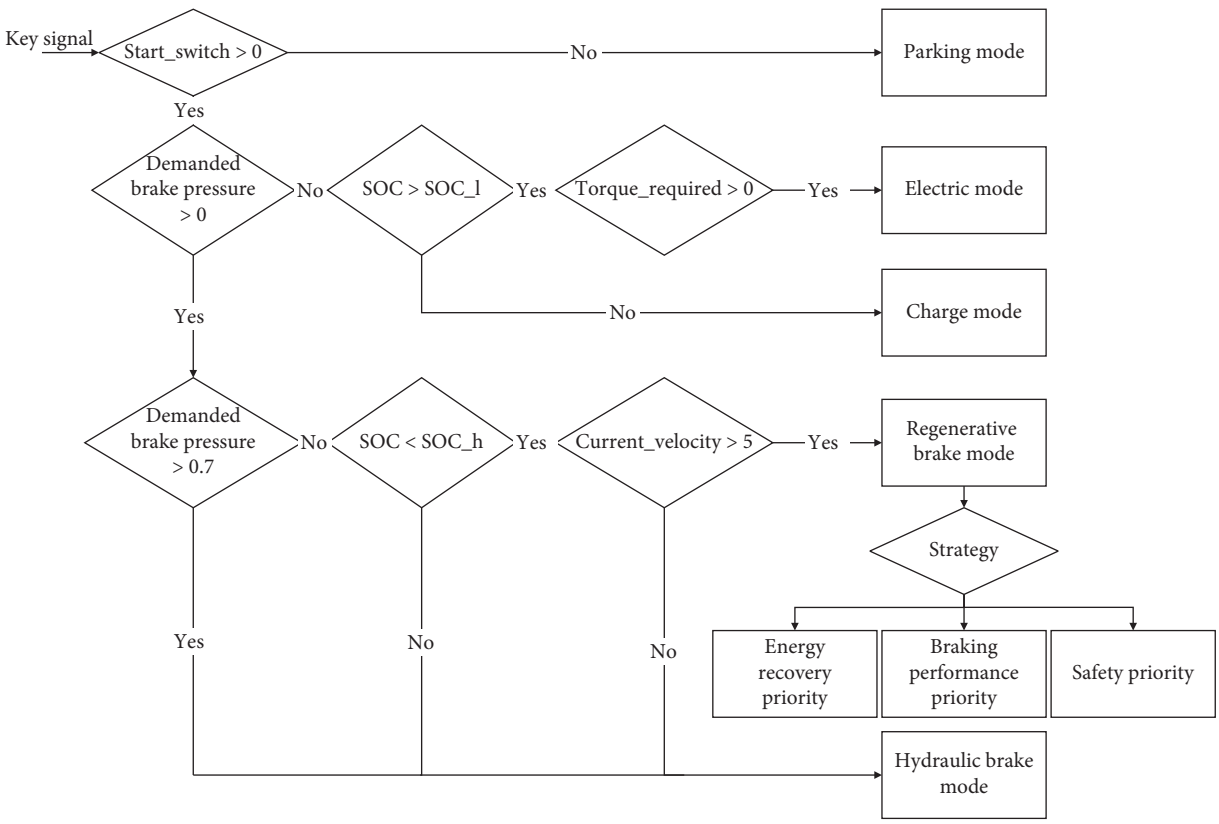


Figure 6: Mode switching logic.

propel the vehicle. Thus, even distribution strategy is defined as the baseline torque split strategy.

In the previous research, another critical strategy is the rule-based torque split strategy, shown in Figure 7. If the demanded driving torque is smaller than a preset torque value T_switch, the demanded driving torque will be totally distributed to a single motor to propel the vehicle. Else, the driving torque is distributed equally to two motors. To define a proper torque threshold T_switch, the optimization procedure is proceeded using the genetic algorithm (GA), in which T_switch is selected as the optimization variable and the energy consumption of typical cycles simulation is chosen as the optimization objective. The optimal T_switch (102.6 Nm) is obtained after 100 iterations. As a contrast, it is also researched in this paper.

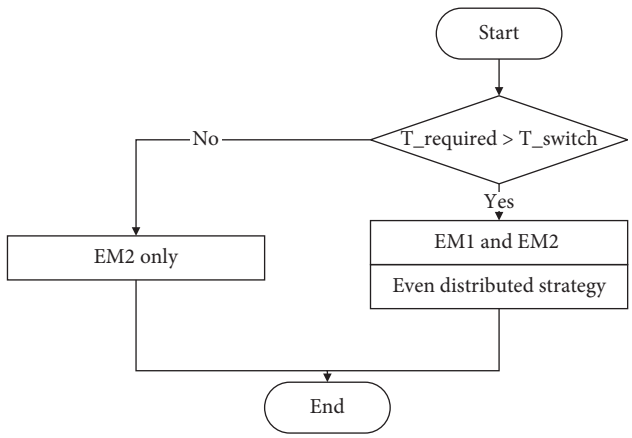


Figure 7: Rule-based torque split strategy.

3.3.2. Real-Time Driving Torque Split Strategy Based on ANLPSO

(1) Optimization problem description

Optimization Variables. The power split ratio is considered as the optimization variable, which is defined as the required power of front motor divided by the total demanded power:

$$\alpha(k) = \frac{P_{\mathrm{EM2}}(k)}{P_d(k)} = \frac{T_{\mathrm{EM2}}(k)n_{\mathrm{EM2}}(k)}{T_{\mathrm{EM1}}(k)n_{\mathrm{EM1}}(k) + T_{\mathrm{EM2}}(k)n_{\mathrm{EM2}}(k)} = \frac{T_{\mathrm{EM2}}(k)}{T_{\mathrm{EM1}}(k) + T_{\mathrm{EM2}}(k)} = \frac{T_{\mathrm{EM2}}(k)}{T_d(k)}, \tag{12}$$

where α is the power split ratio and P_d and T_d are the demanded power and torque of driver, respectively. According to Equation (12), since the powertrain architecture of the front axle is the same as that of rear axle, the motor speed of EM1 is the same as that of EM2. Thus, the power split ratio is the same as the torque split ratio. The optimized output torque of EM1 and EM2 can be given as

$$\begin{aligned} T_{\mathrm{EM2}}(k) &= \alpha(k)T_d(k); \\ T_{\mathrm{EM1}}(k) &= (1-\alpha(k))T_d(k). \end{aligned} \tag{13}$$

According to equations (12) and (13), it is obvious that the range of optimization variable is 0-1. Because of the symmetrical front and rear powertrain architecture, the range of optimization variables can be simplified to 0–0.5.

Constraints. The output torque of EM1 and EM2 is supposed not to exceed the available maximum discharge torque of motor. It is also nonnegative as the energy efficiency is lower when one motor drives the other one to generate electricity. In detail, it can be expressed as

$$\begin{aligned} 0 \le T_{\mathrm{EM1}}(k) &\le T_{\mathrm{dismaxEM1}}(n_{\mathrm{EM1}}(k)), \\ 0 \le T_{\mathrm{EM2}}(k) &\le T_{\mathrm{dismaxEM2}}(n_{\mathrm{EM2}}(k)). \end{aligned} \tag{14}$$

Objective Function. For the given demanded driving torque and motor speed at time k, the main purpose of the optimal driving torque split strategy is to maximize total efficiency of two motors. Hence, the total efficiency is used as objective function, which can be formulated as follows:

$$\eta = \frac{P_{\mathrm{output}}}{P_{\mathrm{in}}} = \frac{T_d}{(\alpha T_d/\eta_{\mathrm{EM2}}) + (1-\alpha)T_d/\eta_{\mathrm{EM1}}} = \frac{1}{\alpha/f(n_{\mathrm{EM2}}, \alpha T_d) + (1-\alpha)/f(n_{\mathrm{EM1}}, (1-\alpha)T_d)}, \tag{15}$$

where η is the total efficiency of two motors, P_{output} is total output power of two motors, and P_{in} is the input power of the powertrain, which is supplied by the battery.

(2) Fundamental principle of adaptive nonlinear particle swarm optimization

Based on the powertrain model in Section 2 and the optimization problem description, it could be found that the optimization of the torque split ratio is a highly nonlinear optimization problem. The optimal torque split ratio can be described as the optimization of α within the simplified range, given the time-varying inputs ($T_d(k)$, $n_{\mathrm{EM1}}(k)$, and $n_{\mathrm{EM2}}(k)$), subject to a set of time-varying constraints ($T_{\mathrm{dismaxEM1}}(k), T_{\mathrm{dismaxEM2}}(k)$).

PSO is a population-based heuristic algorithm introduced by Kennedy and Eberhart in 1995 [29], which mimics the social behavior of animal groups such as a flock of birds or fish shoals guiding particles to find global optimal solution. The optimization procedure can be abstracted as particles flying in the designed N-dimensional space. During each flying process, each particle would update its velocity and position according to previous individual best position and the best position of the swarm. Then, the particle will be compared with the best solution. It

has experienced to update individual best position. And the best position of the swarm is further selected among all individual best positions. The velocity and position of each particle can be calculated using the following equation:

$$\begin{aligned} v_i^N(t+1) &= \xi v_i^N(t) + c_1 r_1\left(\mathrm{pBest}_i^N(t) - p_i^N(t)\right) \\ &\quad + c_2 r_2\left(\mathrm{gBest}^N(t) - p_i^N(t)\right), \\ p_i^N(t+1) &= p_i^N(t) + v_i^N(t+1), \end{aligned} \tag{16}$$

where $v_i^N(t)$ is the current velocity of the ith particle, $i = 1, \ldots,$ sizepop; sizepop is the population size of the swarm; t means the tth iteration, $t = 1, \ldots,$ maxgen; N represents the dimensions of solution space; ξ is the inertia weight, which is used to inherit from previous velocity feature to next iteration; c_1 and c_2 are the learning factors of individual learning and social learning, respectively; r_1 and r_2 are the random number between 0 and 1; $\mathrm{pBest}_i^N(t)$ is the best position of the ith particle in t^{th} iteration; $p_i^N(t)$ is the current position of the ith particle in t^{th} iteration; and $\mathrm{gBest}^N(t)$ is the best position of the whole swarm after t iterations.

It is proved that PSO is a simple, efficient, and robust stochastic optimization method to solve a nonlinear optimization problem using the concept of social interaction. However, the general PSO suffers from some drawbacks, which may result in premature convergence and slow convergence speed [30].

To improve the performance of general PSO, this paper adopts an adaptive nonlinear particle swarm optimization algorithm (ANLPSO). First, in the general PSO, the initial position and velocity of each particle are randomly generated within the designated range (range of the optimization variables). Assuming most of the initial particles centralized in a small region (subset of the designated range), the PSO would be highly possible to get into the local optimum, shown in Figure 8(a). As such, the initialization process is redesigned to guarantee the population diversity in the early optimization process. The designated range is divided into sizepop segments. The initial position of each particle is randomly generated within the range of corresponding segment, shown in Figure 8(b). Thus, the particle is distributed within the designated range as uniformly as possible, as shown in the following equation:

$$p_i^N = p_{\min}^N + \frac{(\mathrm{i}-1)\left(p_{\max}^N - p_{\min}^N\right)}{\text{sizepop}} + \frac{r_3\left(p_{\max}^N - p_{\min}^N\right)}{\text{sizepop}}, \tag{17}$$

where $p_{\min}^N$ and $p_{\max}^N$ are lower bound and upper bound of the designated range, respectively, and r_3 is a random number between 0 and 1.

Second, it is concluded that the bigger ξ value results in the occurrence of exploration behavior, while the smaller ξ value results in the occurrence of exploitation behavior [31]. In the ANLPSO, the inertia weight decreases nonlinearly with the increase in iteration. In the early optimization process, the inertia weight is relatively big to ensure the convergence speed. And the inertia weight decreases with the increasing iteration to improve the quality of solution. When compared to linear decreasing inertia weight, the inertia weight in ANLPSO is a little smaller, resulting in better quality of solution. The inertia weight can be determined as follows:

$$\xi = \xi_{\max} + \left(\frac{t-1}{\text{maxgen}-1}\right)^{0.5}\left(\xi_{\min} - \xi_{\max}\right), \tag{18}$$

where $\xi_{\min}$ and $\xi_{\max}$ are the designated minimum value and maximum value of the inertia weight.

Third, the learning factors c_1 and c_2 are dynamically changing with the fitness value of current particle, shown in equation (19). Particles with above average fitness values can obtain the smaller c_1 value and bigger c_2 value to strengthen their social-learning abilities and improve solutions' qualities. On the contrary, particles with below average fitness values can obtain the bigger c_1 value and smaller c_2 value to strengthen their global search abilities and improve convergence speed:

$$\begin{cases} c_1 = c_{1,\max} - \dfrac{\left(c_{1,\max} - c_{1,\min}\right)\left(\mathrm{Fit}_i - \mathrm{Fit}_{\min}\right)}{\mathrm{Fit}_{\text{avg}} - \mathrm{Fit}_{\min}}, & c_2 = c - c_1, \mathrm{Fit}_i < \mathrm{Fit}_{\text{avg}}, \\ c_1 = c_{1,\min}, c_2 = c - c_1, & \mathrm{Fit}_i \geq \mathrm{Fit}_{\text{avg}}, \end{cases} \tag{19}$$

where $c_{1,\min}$ and $c_{1,\max}$ are minimum and maximum individual-learning factor value, respectively; c is the designated sum value of c_1 and c_2; Fit_i is the fitness value of the ith particle; and $\mathrm{Fit}_{\min}$ and $\mathrm{Fit}_{\text{avg}}$ are the minimum and average fitness value of the whole swarm, respectively.

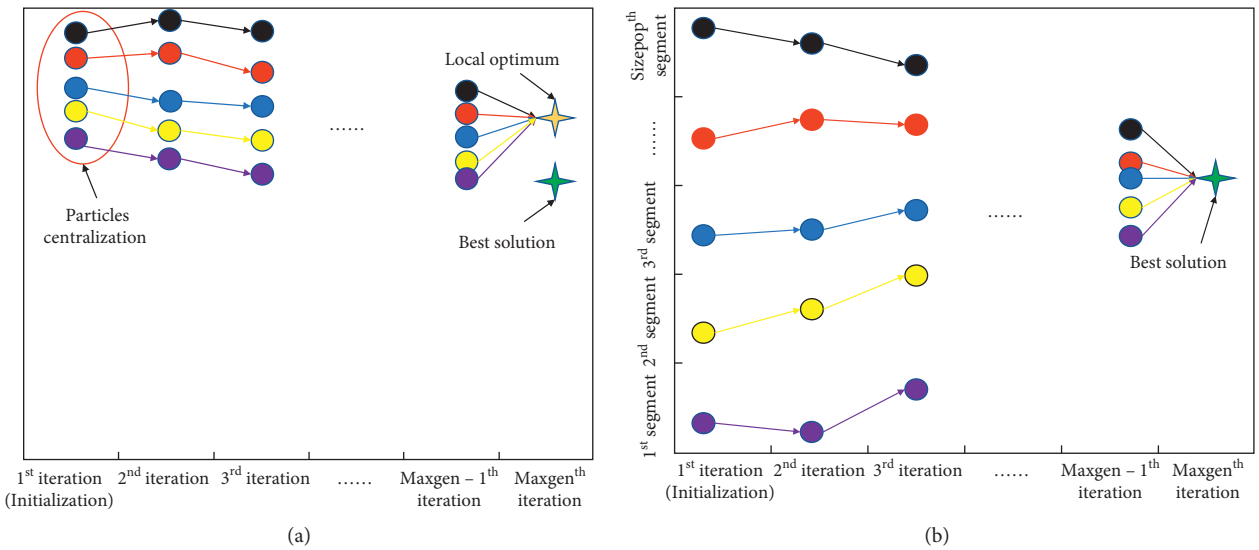


Figure 8: Differences in initialization process between (a) PSO and (b) ANLPSO.

(3) Procedures of adaptive nonlinear particle swarm optimization

To optimize the energy efficiency in this dual-motor vehicle, the ANLPSO algorithm was adopted to control the torque split between two motors. The optimization procedure of the torque split ratio based on ANLPSO (shown in Figure 9) is described as follows [32]:

(1) Adaptive initialization: the initial position of particles is generated according to equation (17) to guarantee the population diversity, while the initial velocity for the particles is randomly generated. The optimization problem in this study is a one-dimensional ($N=1$) problem. Hence, the position and velocity of the particles are described as α_i and v_i, respectively, which are generated as follows:

$$\alpha_i = \alpha_{\min} + \frac{(i-1)(\alpha_{\max}-\alpha_{\min})}{\text{sizepop}} + \frac{r_3(\alpha_{\max}-\alpha_{\min})}{\text{sizepop}},$$
$$v_i = R[v_{\min}, v_{\max}], \tag{20}$$

where $\alpha_{\min}$ and $\alpha_{\max}$ are lower and upper boundaries of torque split ratio, respectively, and $R[v_{\min}, v_{\max}]$ means the random process to generate velocity within the range of minimum velocity and maximum velocity.

(2) Update inertia weight and learning factors: the nonlinear-decreasing inertia weight and dynamic learning factors can be determined according to equations (18) and (19), respectively.

(3) Update velocity and position: the velocity and position of each particle are updated according to equation (16).

(4) Fitness value calculation: as mentioned, the total efficiency is used as objective function of the optimization problem. In ANLPSO, the fitness value of the particle is calculated based on this objective function. And the particle with the maximum fitness value would be selected as the best solution:

$$\text{Fit} = \eta = \frac{1}{\alpha/f(n_{\text{EM2}}, \alpha T_d) + (1-\alpha)/f(n_{\text{EM1}}, (1-\alpha)T_d)}. \tag{21}$$

(5) Update pBest_i and gBest: after the calculation of fitness value, each particle should choose the particle with the maximum fitness value, pBest_i, according to its own experience. Besides, the particle with global best fitness value of the whole swarm should be selected as gBest among all pBest_i. In the first iteration, each particle is directly selected as its own pBest_i.

(6) Stopping rule: Steps (2)–(5) are repeated until the optimization converges or the maximum number of iteration is reached. The best position with the maximum fitness value would be outputted as the optimal torque split ratio of two motors.

4. Results and Discussion

In this study, the simulation was conducted for four cases, including single-motor powertrain, dual-motor powertrain with even distributed strategy, dual-motor powertrain with rule-based strategy, and dual-motor powertrain with ANLPSO strategy. The results would be analyzed with comparison in this section. It is noted that the torque split strategy was not applied into the control strategy for single-motor powertrain. Thus, only the results of energy consumption would be compared with other three cases.

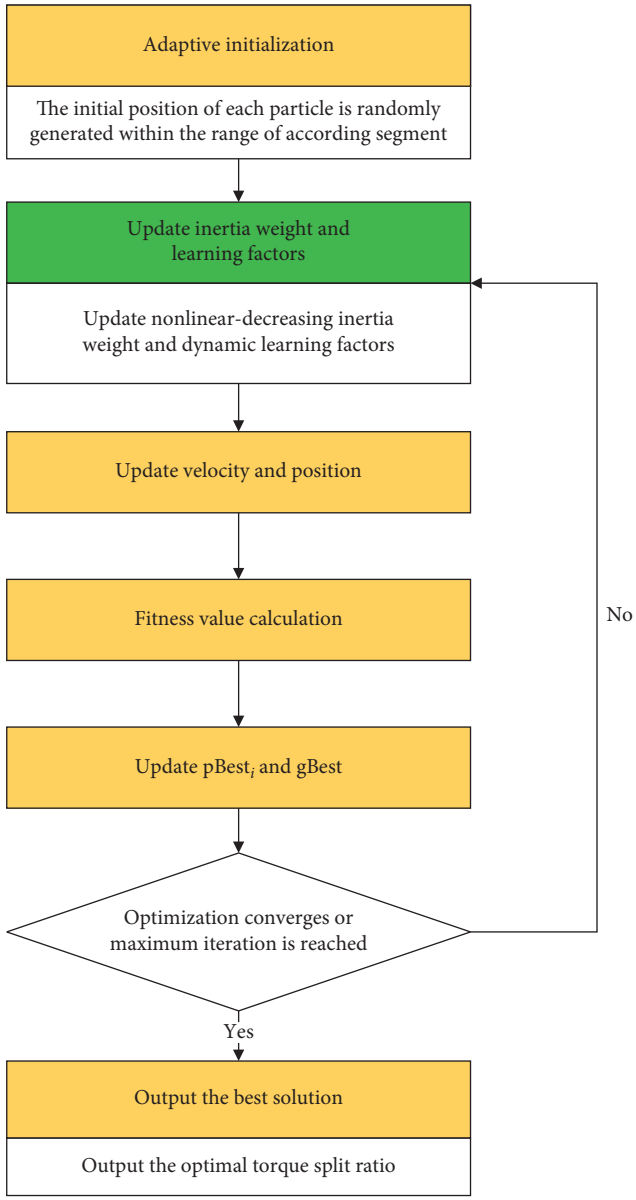


Figure 9: Optimization procedure of ANLPSO.

4.1. Simulation Program and Settings. The simulation was mainly based on Matlab/Simulink platform. The sampling time was set as a fixed sampling time of 0.1 s, and the Bogacki–Shampine solver was selected. Both the driver model, physical vehicle model, and vehicle control unit (VCU) model were conducted, as shown in Figure 10. The ANLPSO strategy was programmed as a S-function block in the torque split layer of VCU. All key parameters of ANLPSO are listed in Table 4.

4.2. Simulation Model Validation. In order to verify the validity and accuracy of the physical simulation model, the real vehicle experiment is conducted. As mentioned in Section 2, the target vehicle is a commercial dual-motor electric vehicle prototype. The powertrain architecture of real vehicle is shown in Figure 11(a). In the development stage, the allocation of driving torque for this sample vehicle is based on the simplest even distributed strategy and the braking energy is not regenerated. Due to the restricted experiment condition, the real-vehicle experiment was conducted on chassis dynamometer using the NEDC (as shown in Figure 11(b)). Both the vehicle velocity and battery SOC results were recorded from VCU through CAN bus acquisition tool and used to validate the corresponding simulation model. It is noted that all parameters and the control strategy of the simulation model are consistent with the experiment.

As shown in Figure 12(a), both the simulation model and experimental vehicle could track the driving cycle well. The vast majority of velocity tracking error of experiment results is within ±5 km/h, while the velocity tracking error of

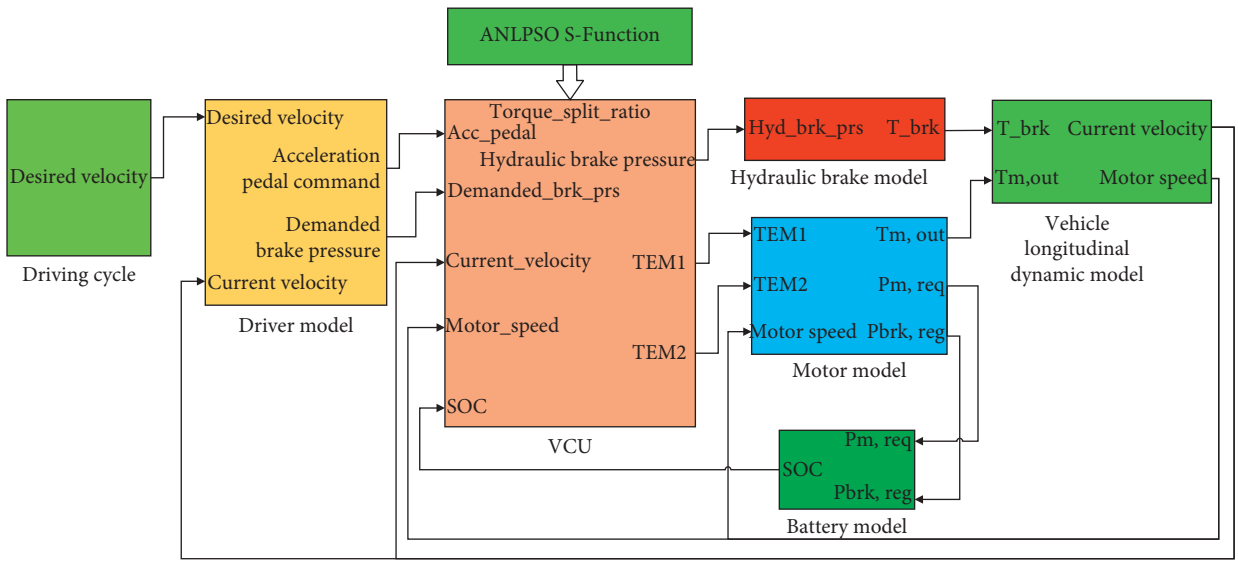


Figure 10: Physical vehicle model with VCU based on Matlab/Simulink platform.

Table 4: Parameters value of ANLPSO.

Parameters	Value
Sizepop	5
Maxgen	20
N	1
$\alpha_{\min}$	0
$\alpha_{\max}$	0.5
$\xi_{\min}$	0.4
$\xi_{\max}$	0.9
$v_{\min}$	−0.25
$v_{\max}$	0.25
$c_{1,\min}$	1.2
$c_{1,\max}$	1.5
c	3

simulation is within ±0.5 km/h. As for the changing tendency of battery SOC, simulation and experimental results are consistent. Because of the sampling accuracy (0.4%) of battery management system, the SOC results of experiment decrease in a ladder shape. The SOC starts with an initial value of 90% and ends with 84.87% and 84.8% in simulation and experiment, respectively, implying a high validity and accuracy of the simulation model, shown in Figure 12(b).

4.3. Simulation Results. With the validated simulation model, comparative simulation results are presented using three different strategies (note that regenerative braking strategy is added to controller in this section). The results in NEDC at the initial SOC 90% are demonstrated in Figures 13 and 14. As illustrated in the first row of Figure 13, the actual velocity (V_c) tracks the desired velocity (V_d) very well for all three cases. The velocity track error is within ±0.4 km/h since more precise driver model based on the feedforward-feedback PI control is adopted, indicating the reliability of the simulation results. The second row presents the torque split ratio (α) results for even distributed strategy. It is

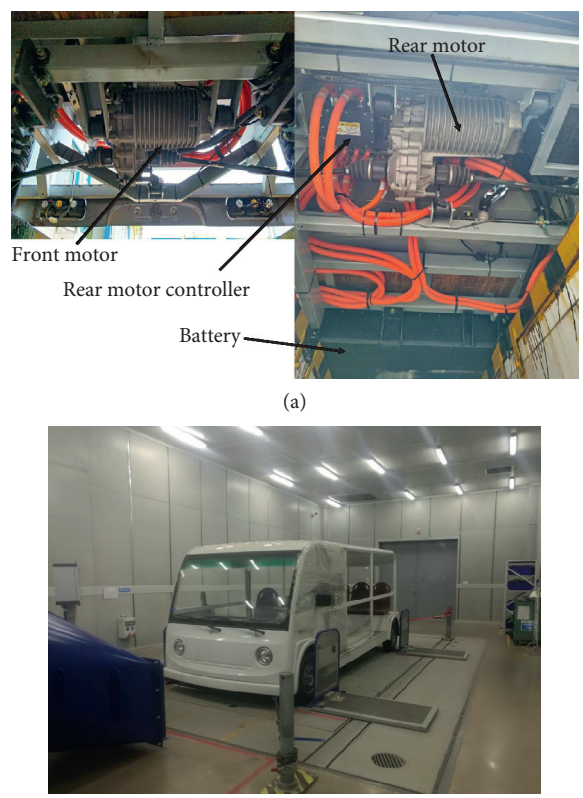


Figure 11: Real-vehicle experiment: (a) powertrain architecture; (b) vehicle experiment on dynamometer.

obvious that the torque split ratio is 0.5 during acceleration and uniform velocity conditions (T_r > 0), which means the desired driving torque is evenly distributed between EM1 and EM2. However, the torque split ratio is 0 in the third row for rule-based strategy, indicating that the EM1 provides driving torque solely in the whole cycle run. In the fourth

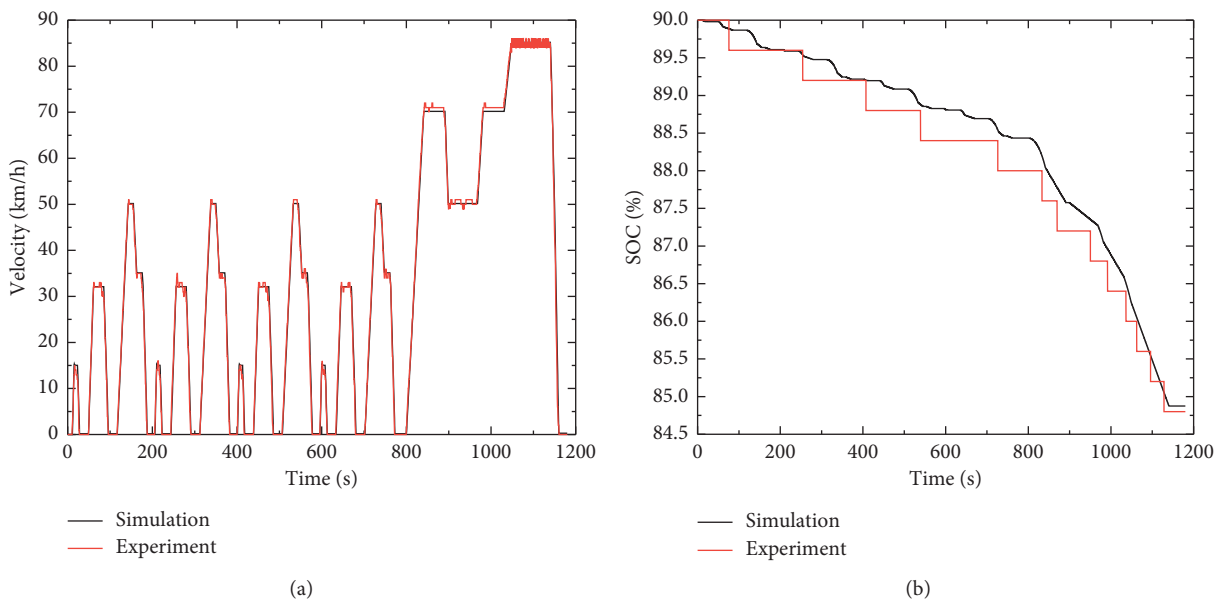


Figure 12: Comparison of simulation and experiment results: (a) driving cycle tracking; (b) battery SOC.

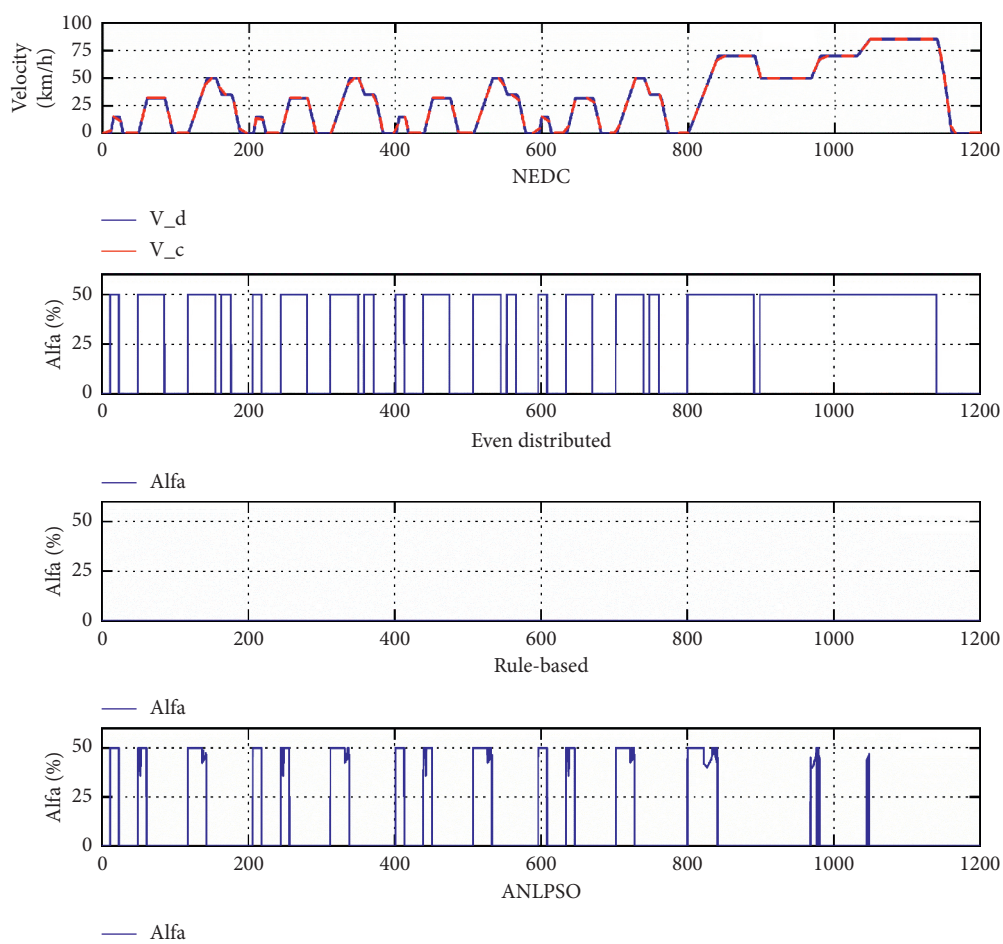


Figure 13: Velocity and driving torque split ratio of three strategies in NEDC.

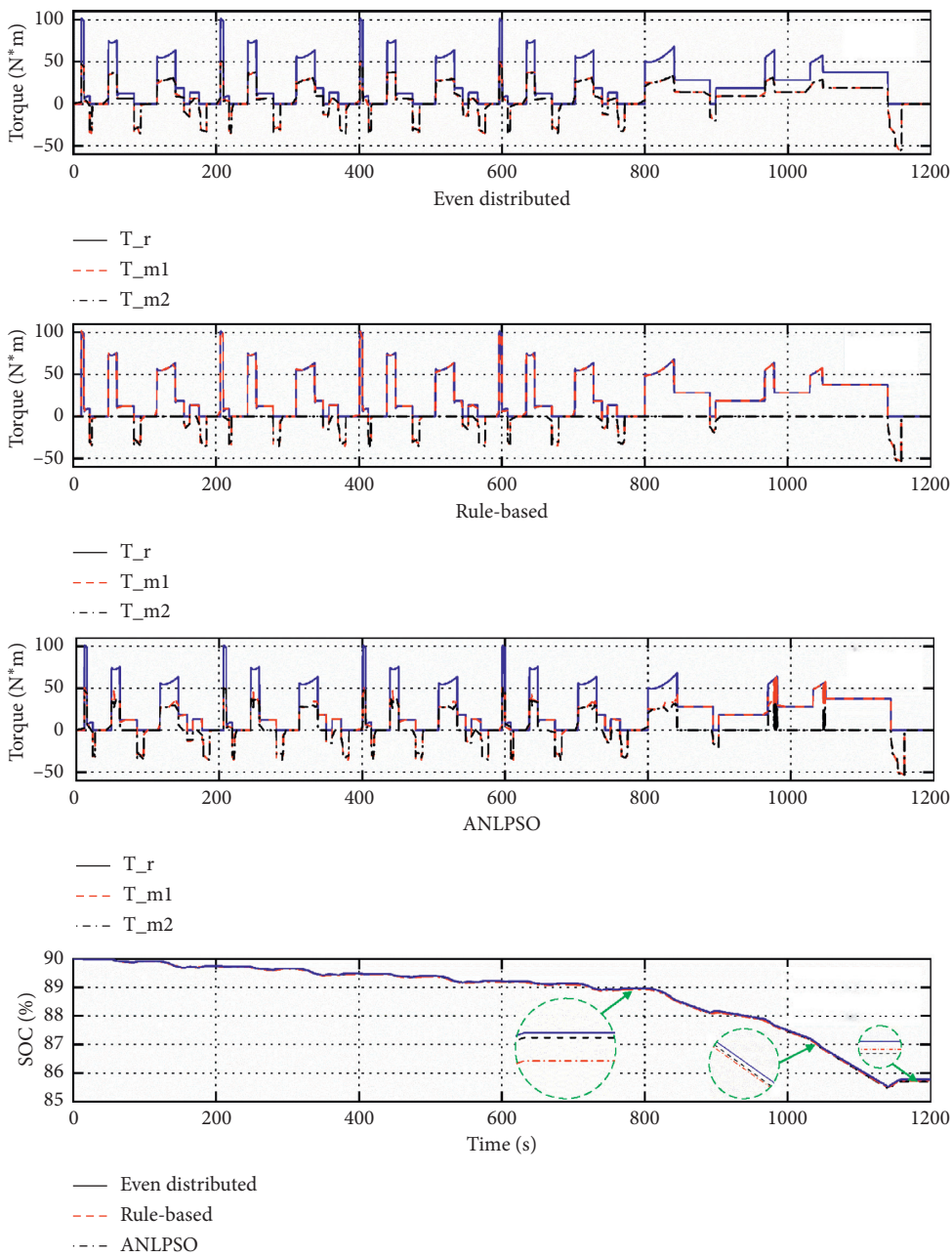


Figure 14: Torque curves and the battery SOC of three strategies in NEDC.

row, the real-time torque split ratio results based on ANLPSO is revealed. Compared with the other two strategies, the optimal torque split ratio changes with torque requirements during acceleration condition, which implies that two motors participate in propulsion more flexibly to achieve a higher total efficiency.

In order to elucidate the difference of those three strategies specifically, the torque results of three cases are illustrated in Figure 14. For the even distributed strategy in the first row, the demanded torque is equally distributed to EM1 and EM2. The torque change trajectory of two motors is exactly the same. As for rule-based strategy, the vehicle is propelled only by the EM1, and the torque change curve of EM1 is the same as that of demanded torque. The torque results are in concert with the torque split ratio results shown in Figure 13. With respect to ANLPSO, the torque of two motors is split based on α. The EM1 predominates in the low torque area (when T_r is small, such as uniform velocity condition), while two motors are both activated to propel the vehicle in high torque area (when T_r is relatively big, such as acceleration condition). Within the high torque area, the torque split ratio is 0.5 mostly. However, it is noticeable that the required torque is not equally distributed to two motors in some of high torque area, which indicates that even distributed strategy is not always the optimal torque split

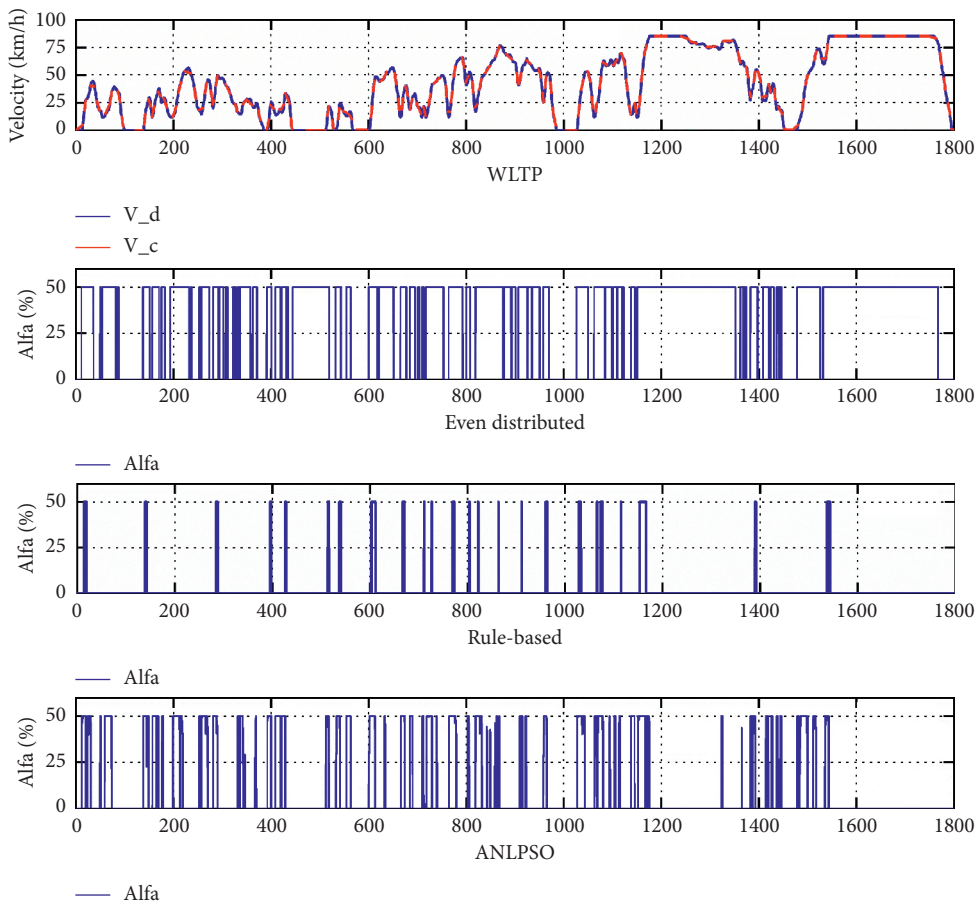


Figure 15: Velocity and driving torque split ratio of three strategies in WLTP.

strategy in high required torque area (in literature review, for dual-motor powertrain with two identical motors, the optimal torque split strategy is even distributed strategy, while the high driving torque is required). The fourth row shows SOC change trajectory of three cases. Since the powertrain is the same, the difference between three cases is small. As can be seen, there are three green dotted circles, which present three zoom-in pictures of SOC change curve in low velocity area, medium velocity area, and high velocity area, respectively. The battery SOC of ANLPSO is always higher than the rest two cases, indicating that the torque split strategy with ANLPSO consumes least energy. The rule-based strategy is least efficient because the required torque is relatively high in most time of first 800s as it is inefficient that single motor drives the vehicle in high torque area. After 800s, most time of cycle run is in the area of uniform velocity, which means the required torque is low. It turns out that single motor is more efficient to supply the power. Furthermore, it can be seen that the difference of battery SOC between even distributed strategy and rule-based strategy becomes smaller and the SOC with rule-based strategy is higher than that with even distributed strategy eventually.

Figures 15 and 16present the simulation results in WLTP. As illustrated in Figure 15, the adaptive driver model still tracks the desired velocity precisely. It is apparent that acceleration and deceleration condition is more compared with NEDC. Consequently, EM2 works more frequently to assist EM1 to propel the vehicle in rule-based strategy, and the optimal torque split ratio based on ANLPSO also changes more frequently to achieve the best energy efficiency. Figure 16 demonstrates the comparison of torque results of three strategies. Similar with the results shown in Figure 14, it is more efficient that single motor supplies the power solely in low torque area, while two motors propel the vehicle together with an appropriate torque split ratio in high torque area. In the fourth row of Figure 16, the results of battery SOC is presented. It is obvious that more electricity is consumed in WLTP relative to in NEDC. In the first 1000s, frequent rapid acceleration indicates a higher required torque. Hence, the battery SOC of rule-based strategy drops more than that of even distributed strategy. Since the best split ratio is not always 0.5 in high torque area, energy efficiency of ANLPSO is still higher than that of even distributed strategy.

After 1100 s, rule-based strategy consumes little energy than even distributed strategy because of the smoother acceleration process and uniform velocity period. And the change tendency of the SOC is similar to the results in the fourth row of Figure 14. In general, ANLPSO exhibits highest energy efficiency while the rule-based strategy is relatively efficient than even distributed strategy in both NEDC and WLTP.

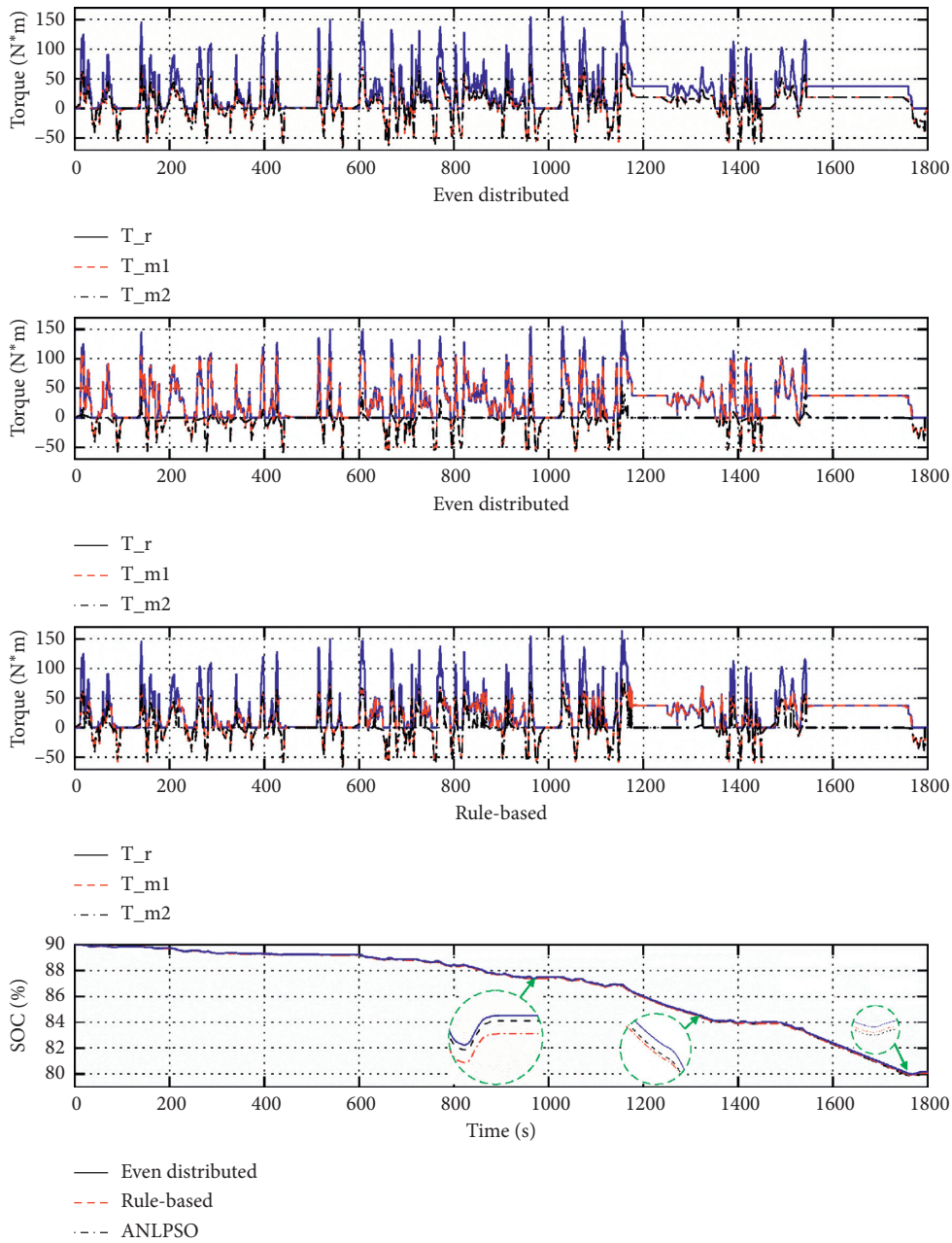


Figure 16: Torque curves and the battery SOC of three strategies in WLTP.

4.4. Optimization Performance and Real-Time Applicability Analysis of ANLPSO. As detailed in Section 3, the torque split ratio is searched based on ANLPSO, which is designed to ameliorate shortcomings of general PSO. To further compare the optimization performance and real-time applicability between general PSO and ANLPSO, the evolution process was investigated. Figure 17 shows the optimization process to search the best torque split ratio at T_r = 85 Nm and Motor_speed = 5400 rpm (to compare two methods fairly, this condition is randomly selected from cycle run). Both the best position of each particle P_{best}, the best position of all particles G_{best}, and the best fitness value in each iteration are demonstrated. As shown in the first row, four particles are randomly generated within a small range (0.35–0.36). As a result, the optimization process is trapped in local optimum at the 7th iteration, and thus, the optimal solution is not found eventually. By contrast, because of the improvement of the initialization process, the nonlinear decreasing inertia weight, and dynamic learning factors, the converge speed and optimization performance of ANLPSO are greatly enhanced. As shown in the second row and the third row, the evolution process is more stable and swifter. The optimal fitness value (fitness value = 93.77345) is found only after 9 iterations, and the best position of each particle P_{best} both converges to the optimal position (α = 0.449) after 16 iterations. Therefore, ANLPSO is verified to possess better

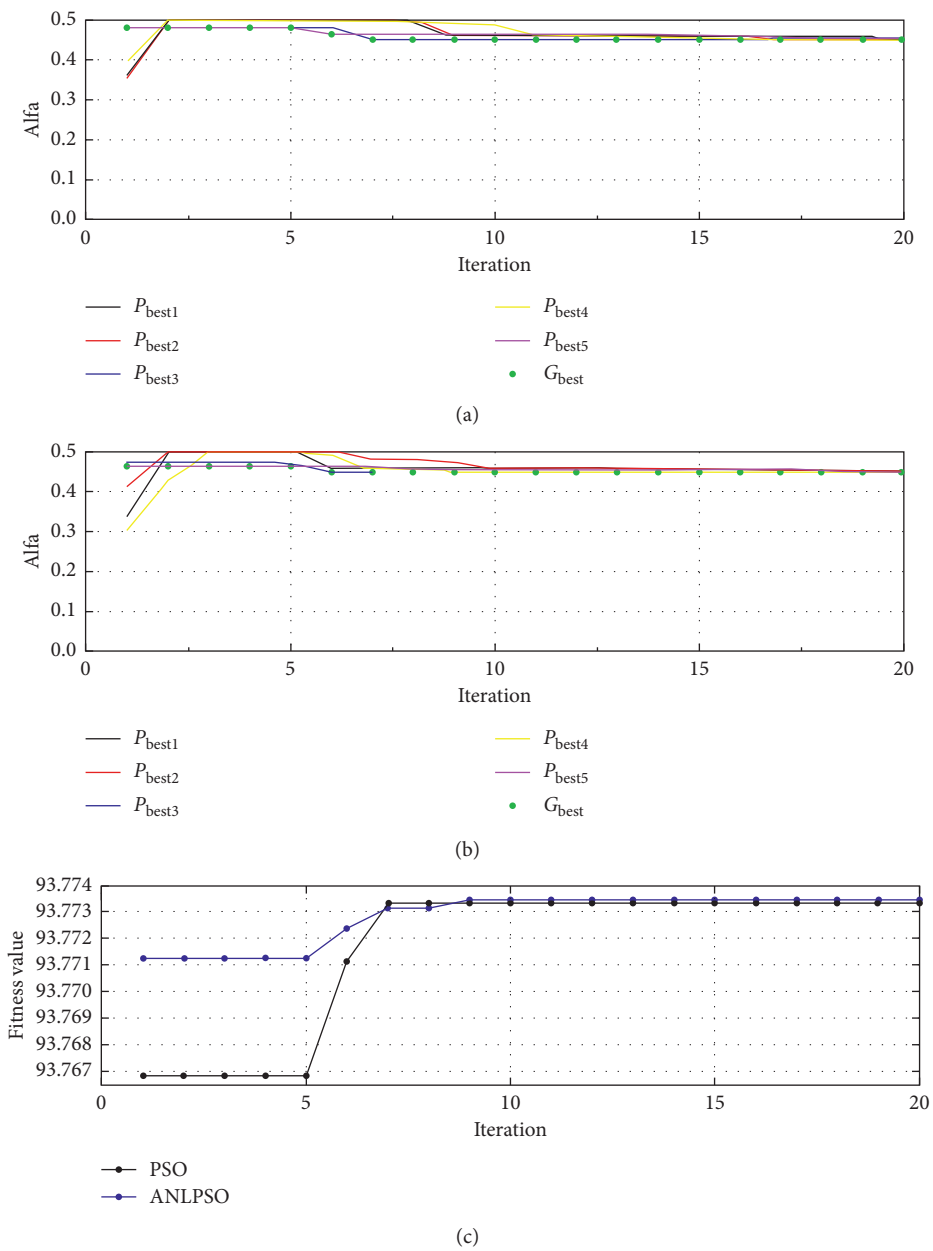


Figure 17: Comparison of convergence performance between PSO and ANLPSO.

convergence performance and global optimization ability than general PSO.

Furthermore, the computational time of the optimal torque split strategies is 377 s and 665 s for NEDC (1180 s) and WLTP (1800 s), respectively. Then, the equivalent computational time per second is approximately 0.031 s and 0.036 s, which shows a great real-time applicability and potential to be implemented in VCU of real vehicle. Before practical applications, to decide the population size and maximum number of iteration, there should be a tradeoff between the optimality and real-time applicability. For example, as for VCU with limited computational capability, it is probably advisable to use a relatively small maximum iteration size and a relatively small population size (10 iterations and 5 populations) to ensure real-time applicability. By contrast, it would be better to determine a large maximum iteration size and a large population size (30 iterations and 10 populations) to take priority of the optimality without sacrifice in real-time implementability.

4.5. Comparison of Efficiency, Motor Loss, and Energy Improvement. To further demonstrate energy-saving potential of dual-motor powertrain over single-motor powertrain and the difference of three control strategies, the total efficiency of powertrain, the power loss of two motors, and energy consumption were investigated. Figure 18 illustrates the results of total efficiency of two motors in NEDC and

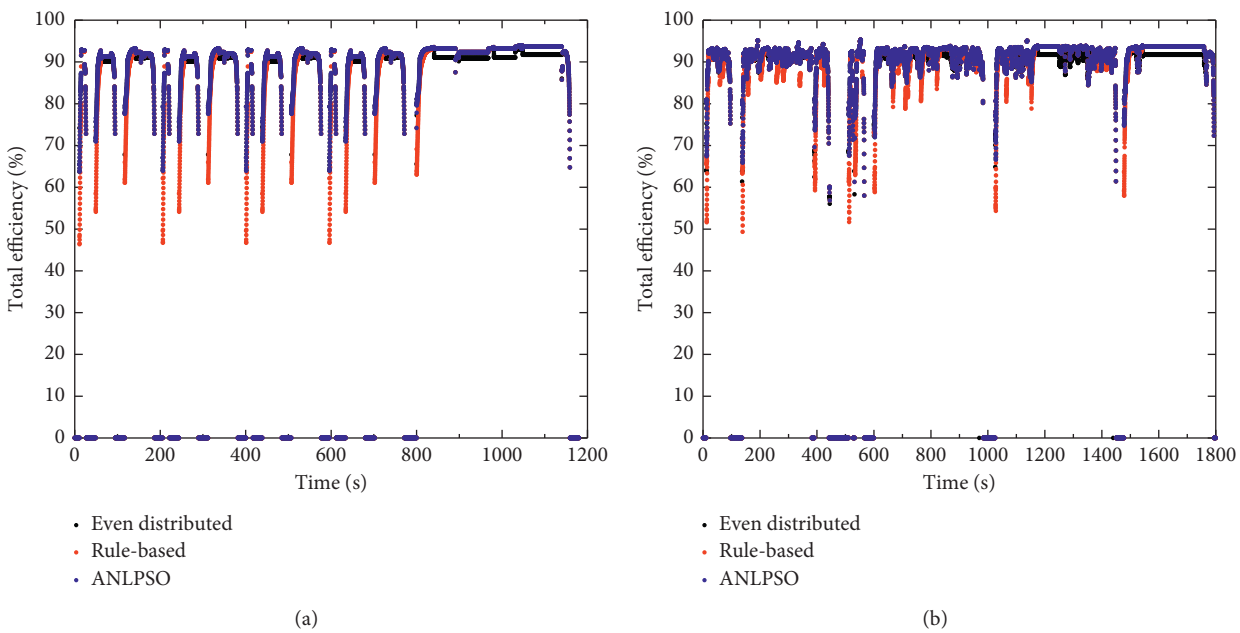


Figure 18: Total efficiency of two motors of three strategies in NEDC and WLTP.

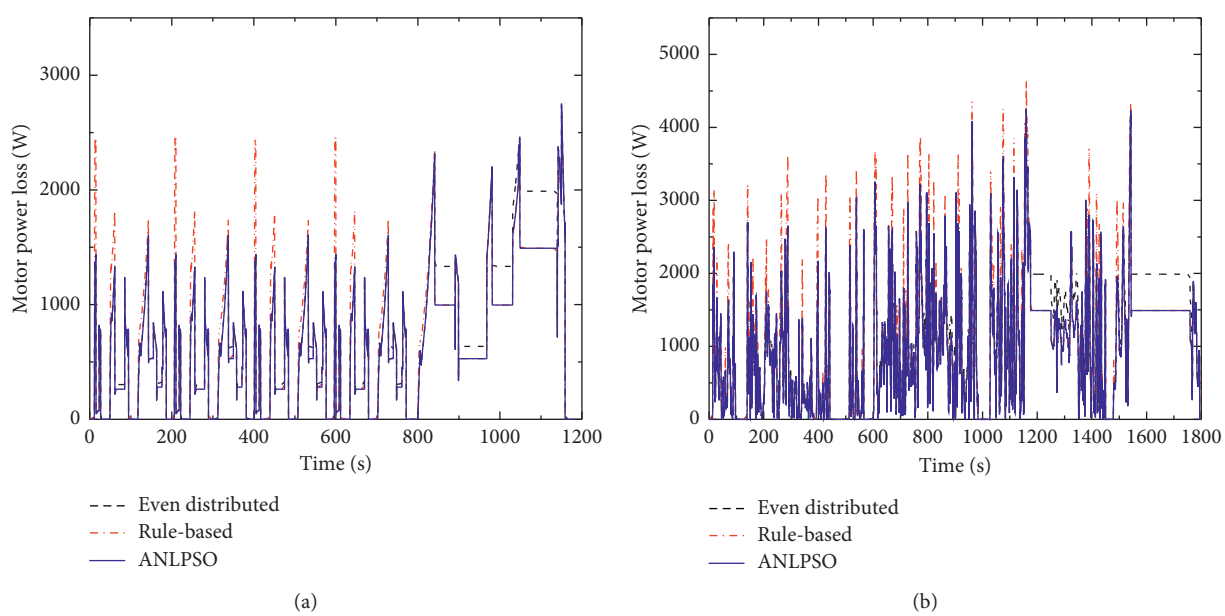


Figure 19: Motor power loss of two motors of three strategies in NEDC and WLTP.

WLTP. The total efficiency of ANLPSO is above 90% in most of cycle run, and even the lowest efficiency is not lower than 65%, which is the highest among three strategies. Consequently, the power loss of two motors is the lowest with ANLPSO, as shown in Figure 19.

As for rule-based strategy, the total efficiency is the lowest in the first 800 s in NEDC and in the first 1000 s in WLTP. However, it becomes more efficient than even distributed strategy when the cycle run enters in low torque area (after 800 s in NEDC and after 1000 s in WLTP). And it is obvious that the motor power loss of rule-based strategy is the most in high torque area and the motor power loss of even distributed strategy is the most in low torque area, as shown in Figure 19. It is worth noting that the total efficiency of even distributed strategy is basically the same as ANLPSO in high torque area, while the total efficiency of rule-based strategy is the same as ANLPSO in low torque area.

The energy consumption is obtained by integrating the battery power in NEDC and WLTP, as listed in Table 5. Assuming that the braking energy of each axle is totally regenerated by the motor mounted on according axle, it is

Table 5: Energy consumption comparison.

	Total motor loss (kWh)		Electricity consumption (kWh/100 km)		Energy saving compared to single-motor powertrain (%)	
	NEDC	WLTP	NEDC	WLTP	NEDC	WLTP
Single-motor powertrain	—	—	20.29	23.16	—	—
Dual-motor powertrain with even distributed strategy	0.2324	0.5330	18.15	20.65	10.55	10.83
Dual-motor powertrain with rule-based strategy	0.2221	0.5107	18.05	20.54	11.04	11.31
Dual-motor powertrain with ANLPSO	0.2041	0.4695	17.88	20.34	11.88	12.18

proved that the energy saving of dual-motor powertrain with three different strategies in NEDC has been improved by around 10.55%, 11.04%, and 11.88%, respectively, while the energy saving improvement of dual-motor powertrain with three different strategies in WLTP is 10.83%, 11.31%, and 12.18%, respectively, compared with single-motor powertrain. In addition, when compared with even distributed strategy and rule-based strategy for dual-motor powertrain, the proposed torque split strategy shows an improvement in total motor loss of 12.17% and 8.1% in NEDC, respectively, and 11.91% and 8.07%, respectively, in WLTP.

5. Conclusion

In this study, a real-time optimal energy management strategy for a dual-motor electric powertrain was proposed. Firstly, the physical vehicle model was implemented in Matlab/Simulink platform. Then, the optimal torque split strategy based on ANLPSO was integrated with VCU by using S-function. Moreover, three cases (single-motor powertrain, dual-motor powertrain with even distributed strategy, and dual-motor powertrain with rule-based strategy) were also modeled for comparison. Several important findings are summarized as follows:

(1) In NEDC and WLTP driving cycles, the dual-motor powertrain with two identical motors could achieve energy saving of (10.55%, 10.83%), (11.04%, 11.34%), and (11.88%, 12.18%) using even distributed strategy, rule-based strategy, and the optimal torque split strategy, compared to single-motor powertrain, respectively. This confirms that the dual-motor powertrain has a great energy efficiency improvement over single-motor powertrain.

(2) With respect to dual-motor powertrain, compared to even distributed strategy, the proposed optimal torque split strategy could reduce the total motor loss by 12.17% and 11.91% in NEDC and WLTP, respectively. And the improvement over rule-based strategy is 8.1% and 8.07% accordingly.

(3) As for the ANLPSO, the real-time best torque split ratio could be searched swiftly with three inputs (motor speed, the required torque, and the available maximum torque of each motor) and fitness function of total efficiency of two motors. It is also shown that it has better convergence speed and global optimization performance than conventional PSO, which is provably real-time available in a real controller to achieve the optimal torque allocation.

(4) Based on the detailed result analysis of three strategies, it is concluded that, for dual-motor with two identical motors, the single motor is more efficient to provide the power in low torque area. However, within high torque area, both two motors should be activated to propel the vehicle with an appropriate torque split ratio. In most cases, the optimal torque split ratio is 0.5, which means the torque should be equally distributed to two motors. However, 0.5 is not always the best split ratio according to searching results of ANLPSO. Hence, a proper methodology to figure out the real-time optimal torque split ratio is essential to manage the torque distribution of dual motor powertrain, which is also the main contribution of this study.

Our future work could focus on the applicability of the optimal torque split strategy in the real vehicle test. Furthermore, the research would be expanded into the integrated optimization framework for component sizing and optimal energy management strategy for dual-motor powertrain.

Data Availability

The following data, models, or results that support the findings of this study are available from the corresponding author upon reasonable request: (1) vehicle parameters; (2) motor efficiency data; (3) vehicle physical model; and (4) results data.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The authors would like to express their gratitude to Tsinghua University under project 2017YFB0103902, Wuhan University of Technology under the 111 project (no. B17034), and Innovative Research Team Development Program of Ministry of Education of China (no. IRT_17R83) for supporting this study.

Supplementary Materials

In the "experiment results" folder, the excel file named "experiment results" is the results from real-vehicle experiment validation, while the excel file named "simulation model" is the results of the validated simulation model. These results are mainly for simulation model validation. In the "simulation results" folder, there are two folders named "NEDC" and "WLTP," respectively, which present the simulation results in NEDC and WLTP, respectively. The excel file named "EVEN" is the results of even distributed strategy. The excel file named "Rule-based" is the results of the rule-based strategy. The excel file named "ANLPSO" is the results of the optimal torque split strategy using ANL-PSO. These results are mainly for comparison between different torque split strategies. (*Supplementary Materials*)

References

[1] L. C. A. Silva, J. J. Eckert, F. M. Santiciolli, E. S. Costa, F. G. Dedini, and F. C. Correa, "A study of battery power for a different electric vehicle propulsion system," in *Proceedings of the 2015 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles (ESARS)*, pp. 1–6, IEEE, Aachen, Germany, March 2015.

[2] T. Wilberforce, Z. El-Hassan, F. N. Khatib et al., "Developments of electric cars and fuel cell hydrogen electric cars," *International Journal of Hydrogen Energy*, vol. 42, no. 40, pp. 25695–25734, 2017.

[3] K. T. Chau and C. C. Chan, "Emerging energy-efficient technologies for hybrid electric vehicles," *Proceedings of the IEEE*, vol. 95, no. 4, pp. 821–835, 2007.

[4] G. Wu, X. Zhang, and Z. Dong, "Powertrain architectures of electrified vehicles: review, classification and comparison," *Journal of the Franklin Institute*, vol. 352, no. 2, pp. 425–448, 2015.

[5] H. Eckard and P. Marx, "Electric cars: technical characteristics and environmental impacts," *Environmental Sciences Europe*, vol. 24, no. 1, pp. 1–15, 2012.

[6] O. van Vliet, A. S. Brouwer, T. Kuramochi, M. van den Broek, and A. Faaij, "Energy use, cost and CO_2 emissions of electric cars," *Journal of Power Sources*, vol. 196, no. 4, pp. 2298–2310, 2011.

[7] S. Chopra and P. Bauer, "Driving range extension of EV with on-road contactless power transfer: a case study," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 1, pp. 329–338, 2013.

[8] M. A. Miller, A. G. Holmes, B. M. Conlon, and P. J. Savagian, "The GM "voltec" 4ET50 multi-mode electric transaxle," *SAE International Journal of Engines*, vol. 4, no. 1, pp. 1102–1114, 2011.

[9] F. Di Nicola, A. Sorniotti, T. Holdstock, F. Viotto, and S. Bertolotto, "Optimization of a multiple-speed transmission for downsizing the motor of a fully electric vehicle," *SAE International Journal of Alternative Powertrains*, vol. 1, no. 1, pp. 134–143, 2012.

[10] Q. Ren, D. A. Crolla, and A. Morris, "Effect of transmission design on electric vehicle (EV) performance," in *Proceedings of the 2009 IEEE Vehicle Power and Propulsion Conference*, pp. 1260–1265, IEEE, Dearborn, MI, USA, September 2009.

[11] J. Ruan, P. Walker, and N. Zhang, "A comparative study energy consumption and costs of battery electric vehicle transmissions," *Applied Energy*, vol. 165, pp. 119–134, 2016.

[12] F. C. Correa, J. J. Eckert, L. C. A. Silva, F. M. Santiciolli, E. S. Costa, and F. G. Dedini, "Study of different electric vehicle propulsion system configurations," in *Proceedings of the 2015 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–6, Montreal, Quebec, Canada, October 2015.

[13] J. J. Eckert, L. C. A. Silva, E. S. Costa, F. M. Santiciolli, F. G. Dedini, and F. C. Corrêa, "Electric vehicle drivetrain optimisation," *IET Electrical Systems in Transportation*, vol. 7, no. 1, pp. 32–40, 2017.

[14] H. Qian, G. Xu, J. Yan, T. L. Lam, Y. Xu, and K. Xu, "Energy management for four-wheel independent driving vehicle," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5532–5537, IEEE, Taipei, Taiwan, 2010.

[15] V. Ivanov, D. Savitski, and B. Shyrokau, "A survey of traction control and antilock braking systems of full electric vehicles with individually controlled electric motors," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 3878–3896, 2015.

[16] Y. Chen, X. Li, C. Wiet, and J. Wang, "Energy management and driving strategy for in-wheel motor electric ground vehicles with terrain profile preview," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 3, pp. 1938–1947, 2014.

[17] M. Hu, J. Zeng, S. Xu, C. Fu, and D. Qin, "Efficiency study of a dual-motor coupling EV powertrain," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2252–2260, 2015.

[18] Y. Gao, W. Wang, and Y. Li, "Optimization of control strategy for dual-motor coupling propulsion system based on dynamic programming method," in *Proceedings of the 2019 3rd Conference on Vehicle Control and Intelligence (CVCI)*, pp. 1–6, Hefei, China, September 2019.

[19] B. Wang, M. Xu, and L. Yang, "Study on the economic and environmental benefits of different EV powertrain topologies," *Energy Conversion and Management*, vol. 86, pp. 916–926, 2014.

[20] J. Ruan and Q. Song, "A novel dual-motor two-speed direct drive battery electric vehicle drivetrain," *IEEE Access*, vol. 7, pp. 54330–54342, 2019.

[21] J. Kang, J. Yoo, and K. Yi, "Driving control algorithm for maneuverability, lateral stability, and rollover prevention of 4WD electric vehicles with independently driven front and rear wheels," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 2987–3001, 2011.

[22] P. D. U. Coronado and H. Ahuett-Garza, "Control strategy for power distribution in dual motor propulsion system for electric vehicles," *Mathematical Problems in Engineering*, vol. 2015, Article ID 814307, 10 pages, 2015.

[23] B. Sun, S. Gao, C. Ma, and J. Li, "System power loss optimization of electric vehicle driven by front and rear induction motors," *International Journal of Automotive Technology*, vol. 19, no. 1, pp. 121–134, 2018.

[24] X. Yuan and J. Wang, "Torque distribution strategy for a front- and rear-wheel-driven electric vehicle," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3365–3374, 2012.

[25] D. Wang and B. Wang, "Research on driving force optimal distribution and fuzzy decision control system for a dual-motor electric vehicle," in *Proceedings of the 2015 34th Chinese Control Conference (CCC)*, pp. 8146–8153, Chinese Association of Automation, Hangzhou, China, July 2015.

[26] M. Zhao, J. Shi, and C. Lin, "Energy management strategy design for dual-motor coaxial coupling propulsion electric city-buses," *Energy Procedia*, vol. 152, pp. 568–573, 2018.

[27] P. Shen, Z. Zhao, X. Zhan, and J. Li, "Particle swarm optimization of driving torque demand decision based on fuel economy for plug-in hybrid electric vehicle," *Energy*, vol. 123, pp. 89–107, 2017.

[28] J. Ruan, P. D. Walker, P. A. Watterson, and N. Zhang, "The dynamic performance and economic benefit of a blended braking system in a multi-speed battery electric vehicle," *Applied Energy*, vol. 183, pp. 1240–1258, 2016.

[29] S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury, "Simulation optimization: a review of algorithms and applications," *Annals of Operations Research*, vol. 240, no. 1, pp. 351–380, 2016.

[30] H. S. Ramadan, A. F. Bendary, and S. Nagy, "Particle swarm optimization algorithm for capacitor allocation problem in distribution systems with wind turbine generators," *International Journal of Electrical Power & Energy Systems*, vol. 84, pp. 143–152, 2017.

[31] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, "A comprehensive review of swarm optimization algorithms," *PLoS One*, vol. 10, no. 5, 2015.

[32] S.-Y. Chen, Y.-H. Hung, C.-H. Wu, and S.-T. Huang, "Optimal energy management of a hybrid electric powertrain system using improved particle swarm optimization," *Applied Energy*, vol. 160, pp. 132–145, 2015.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 3697625, 13 pages
https://doi.org/10.1155/2020/3697625

Research Article

Short-Term Traffic Speed Prediction Method for Urban Road Sections Based on Wavelet Transform and Gated Recurrent Unit

Xin Fu,[1,2] Wei Luo,[1] Chengyao Xu,[1] and Xiaoxuan Zhao[1]

[1]*College of Transportation Engineering, Chang'an University, Xi'an 710064, China*
[2]*Engineering Research Center of Highway Infrastructure Digitalization, Ministry of Education, Xi'an, Shaanxi 710064, China*

Correspondence should be addressed to Wei Luo; 2018123030@chd.edu.cn

Received 23 January 2020; Revised 20 March 2020; Accepted 7 April 2020; Published 20 May 2020

Guest Editor: Feng-Jang Hwang

As a core component of the urban intelligent transportation system, traffic prediction is significant for urban traffic control and guidance. However, it is challenging to achieve accurate traffic prediction due to the complex spatiotemporal correlation of traffic data. A road section speed prediction model based on wavelet transform and neural network is, therefore, proposed in this article to improve traffic prediction methods. The wavelet transform is used to decompose the original traffic speed data, and then the coefficients obtained after the decomposition are used to reconstruct the high-frequency random sequences and the low-frequency trend sequence. Secondly, a GRU neural network is constructed to learn the trend of low-frequency sequence. The spatiotemporal correlation between input data is extracted by adjusting the input of the model. Meanwhile, an ARMA model is used to fit unstable random fluctuations of high-frequency sequences. Last of all, the prediction results of the two models are added together to obtain the final prediction result. The proposed prediction model is validated by using road section speed data based on the floating car data collected in Ningbo. The results show that the proposed model has high accuracy and robustness.

1. Introduction

With the socioeconomic development and the acceleration of urbanization, the demand for transportation continues to grow. Though the urban transportation system has been developing as well, trying to match with the increasing demand for transportation, and its supply capacity is improving through the construction of transportation infrastructure, it is still one step behind. Congestion has become a universal problem, causing headache all over the word. Problems brought in by the overcrowding of urban roads include economic, health, and environmental problems, such as stress, fuel consumption, wasted time, and traffic accidents [1]. The intelligent transportation system (ITS) can effectively control and induce urban traffic. It is a key way to solve urban traffic problems, so it puts forward higher requirements for accurate intelligent traffic services [2]. The objects of traffic prediction are traffic parameters such as vehicle flow, speed, and occupancy in a specific area and time period [3]. High-precision traffic prediction can provide accurate travel information for urban residents, and it helps to realize ITS applications [4]. Real-time traffic data can be obtained effectively with the development of ITS [5]. Therefore, in order to improve the accuracy and reliability of traffic prediction, researchers are committed to develop and improve effective traffic prediction models based on fully mined historical traffic data. In this paper, traffic speed prediction on road sections is studied using historical traffic speed data.

It is the research focus of scholars to consider the spatial correlation between road sections in traffic prediction models due to the complex spatiotemporal correlation between road sections. The traffic conditions of each section of the road network are often affected by the conditions of its upstream and downstream sections. For example, traffic congestion often starts on one or more sections and spreads to other sections after a period of time, resulting in regional congestion [6]. Regarding this trait of congestion, some scholars in early years had built nonparametric models using speed data of the studied road section and its upstream and

downstream sections, which can better capture the spatiotemporal correlation between road sections and thus will improve the accuracy of the prediction models [7].

Due to their high flexibility, good learning, and generalization capabilities, algorithms based on neural networks have been widely used in transportation-related tasks [8]. Recurrent neural network (RNN) is applied to traffic prediction because of its special internal structure capable of effectively processing time series. RNN used for traffic prediction mainly includes Long Short-Term Memory (LSTM) neural networks and Gated Recurrent Unit (GRU). GRU was proposed by Cho et al. in 2014 and it is also reported that GRU achieves equal to or better performance than LSTM [9]. In addition, it is proven that GRU outperformed LSTM on nearly all tasks except language modelling with the naive initialization [10]. Dai et al. used GRU network to make short-term traffic forecasts. In the study, GRU was used to process the spatiotemporal feature information of the internal traffic flow of the matrix to achieve the purpose of prediction [11]. In another hybrid model which predicts lane speed, a GRU network layer was used to achieve the final speed prediction [12]. These studies have shown that GRU is competent on traffic prediction and achieve good results.

At the same time, the traffic speed data generated by each road section also has complex attributes. Firstly, the real-time and dynamic nature of urban traffic results in strong random fluctuations in traffic speed data. The difficulty of traffic prediction is enhanced by this inherent characteristic. Secondly, urban traffic has certain spatiotemporal characteristics and periodic laws, which makes the traffic data of each road section have a relatively constant change trend. For example, commuting sections have low traffic speed during morning and evening rush hours and have high traffic speed during other hours. This makes the trend of traffic data over time more predictable. In order to effectively learn the stable periodic characteristics of traffic data and random fluctuations under real-time dynamic traffic, wavelet analysis theory is applied to perform traffic prediction. The historical traffic data are decomposed into subsequences from high to low in terms of frequencies by wavelet transform (WT). The low-frequency sequence contains characteristics similar to the original data. From a long-term perspective, the volatility trend of low-frequency sequence has a repeating daily periodicity. From a short-term perspective, the data at the moments before and after the sequence are similar and continuous [13]. In the past, some scholars used different models to predict the sequence after WT, including ARIMA model and BP neural network [14–16]. However, most of the inputs of these combined prediction models are single time series, which ignores the correlation between the traffic data generated by spatially adjacent road sections.

Therefore, a hybrid prediction framework has been proposed to make more accurate predictions in this paper. Firstly, the historical traffic speed data are decomposed into subsequences from high to low in terms of frequencies by WT. Moreover, a GRU network is constructed to learn the development trend of low-frequency sequence. By controlling the input of this model, the spatiotemporal correlation of traffic speed data can be extracted effectively. At the same time, an autoregressive moving average (ARMA) model is used to fit the random fluctuations on the high-frequency sequences. Finally, the prediction results of the two models are added together to represent the final prediction result.

The rest of the paper was organized as follows. Previous research studies are discussed in Section 2. And the prediction model proposed will be explained by us in Section 3. In Section 4, the validity and robustness of the model are proven using the speed data set in Ningbo, China. The conclusions and future work of this paper are summarized in the end.

2. Related Work

While long-term prediction predicts the future traffic demand using data such as socioeconomic attributes, the traffic prediction required by ITS is a short-term traffic prediction which mainly focuses on the traffic conditions in next few minutes to several hours [1]. In the past few decades, a series of studies on traffic prediction have been implemented by researchers. The existing methods can be divided into three categories: parametric methods, nonparametric methods, and hybrid model prediction methods.

The structure of prediction model based on the parameter methods is rather simple. Parametric methods such as Autoregressive Integrated Moving Average (ARIMA) [17, 18] and Kalman filtering [19, 20] achieve promising results but they rely on certain physical or statistical assumptions. However, traffic flow has the characteristics of randomness and nonlinearity, and it is difficult to establish a parameter model that can reproduce the characteristics of traffic flow in practice. The gradual popularization of sensors on urban roads and GPS on vehicles enabled the acquisition of real-time traffic data. Therefore, the nonparameter method of modelling using large amount of historical data is widely applied. Nonparametric models mainly include the most traditional statistical machine learning methods and the most popular artificial intelligence algorithms. Support vector machine (SVM) [21, 22], K-nearest neighbor (KNN) [23, 24], and artificial neural network (ANN) are the most widely used ones [25].

Intelligent algorithms have attracted widespread attention not only in academia and industry but also in the field of transportation in recent years. RNN is widely recognized as a suitable method to capture the temporal evolution of traffic flow. LSTM, as a typical representative of RNN, was used by many scholars for short-term traffic prediction. Ma et al. applied LSTM to road traffic speed and volume prediction for the first time. By introducing the forget gate, the network can internally connect time series data and increase prediction accuracy [26]. Experimental results show that this network model is superior to ordinary neural networks. The variant GRU of LSTM, which was proposed in 2014, has also been used in traffic prediction because of its simpler structure and similar effects with LSTM [11, 12]. Convolutional neural networks (CNN) algorithm is widely used

in computer vision and image classification [27]. In 2017, CNN was proven to be suitable for traffic speed prediction of the entire road network by learning traffic as images [28]. However, due to the complexity of the topological structure of urban road networks, it is difficult for traditional CNN networks to obtain the spatial characteristics of irregular grid structures. A model called graph convolutional network (GCN) is used to extract the spatial correlation between the road sections. For example, a spatiotemporal GCN (STGCN) model was proposed to extract the spatiotemporal dependence of road network traffic speed data and make predictions [29]. In addition, a model called diffusion convolutional CNN (DCRNN), which combines GCN and RNN at the same time, models the spatial dependence of traffic as a diffusion process on a directed graph and uses RNN to fit temporal correlation [30]. An unsupervised algorithm named sparse autoencoder (SAE) was firstly used to identify and predict traffic state [31]. Furthermore, a deep belief network (DBN) trained with a greedy unsupervised method is also used to predict the traffic speed of an arterial in Beijing [32]. Most neural network models do not provide a reasonable explanation, which is different from statistical machine learning algorithms. However, neural network models have higher prediction accuracy, especially when dealing with large amounts of data.

Short-term traffic prediction can be affected by many factors due to different prediction scenarios. A single prediction model may not be suitable for all scenarios. Fusco et al. built a two-layer model combining Bayesian network and neural network and verified it with floating vehicle data [33]. There are also scholars who combined unsupervised learning algorithms with supervised learning algorithms for model. A prediction model combining SAEs and LSTM is proposed, which uses SAEs and LSTM to extract the spatial and temporal correlations of traffic data, respectively [34]. In the research of Duan, CNN for extracting spatial features and LSTM for capturing temporal information were combined [35]. The method based on wavelet transform has also been applied to predicted the traffic. After the data is decomposed and reconstructed by WT, the neural network optimized by particle swarm is used to predict the sequence separately [36]. A speed prediction study down to the lane scale has also been proposed recently. The researchers built a two-layer deep learning framework that combines LSTM and GRU to predict the speed of different lanes [12]. The combined prediction model combines different prediction algorithms, which can give full play to the advantages of each model to obtain better predicted results.

3. Methodology

3.1. Framework Overview. A W-GRU-ARMA model was constructed for short-term traffic speed prediction in this section. As its name implies, the model is composed of three parts: wavelet transform (W), GRU, and ARMA. The spatiotemporal relationship of urban traffic speed data is taken into account in this model. It focuses on the short-term correlation of traffic speed data for the predicted road sections in the time dimension and the spatial correlation with the upstream and downstream sections. Figure 1 shows the overall architecture. After the wavelet transform of the traffic speed time series data, different low-frequency and high-frequency sequences will be obtained. Therefore, a GRU model for predicting low-frequency trend sequence and an ARMA model for predicting high-frequency random sequences are established. The final prediction result is obtained by summing the prediction results of each model.

The input of the constructed GRU prediction model contains the traffic speed data for predicting road sections and its upstream and downstream sections. For example, the road section A is selected as a research object, and the speed of the road section A at $t+1$ is used as a prediction target. Then, the input vector X of the GRU model is as follows:

$$X = \left[x_{i+1,t-m}, \ldots, x_{i+1,t-1}, x_{i+1,t}, x_{i,t-m}, \ldots, x_{i,t-1}, x_{i,t}, x_{i-1,t-m}, \ldots, x_{i-1,t-1}, x_{i-1,t}\right], \tag{1}$$

where i represents the predicted road section, $i+1$ and $i-1$ represent the downstream and upstream sections of the predicted road section, respectively, t represents current time, and $t-m$ represents the m previous time. The value of m is determined according to the algorithm performance during the model training phase.

3.2. Wavelet Transform. Wavelet transform is a method for processing nonstationary and nonlinear signals with the advantages of multiresolution and multiscale. Wavelets provide an output in terms of the time-frequency scale, which can approach the original signal at any scale and capture the details of the original signal [37]. In nonstationary data analysis, wavelet transform is often used to extract the trend information of sequence changes [38]. The discrete wavelet transform (DWT) can decompose the original traffic speed data into a series of multiple frequencies. The Mallat algorithm is efficient in nonstationary traffic speed time series into sequences of different frequencies through high-pass filters and low-pass filters. The outputs of the low-pass filter and the high-pass filter are defined by dA and dD in equations (2) and (3) which are called approximate coefficients and detail coefficients:

$$dA = \sum_{k=-\infty}^{\infty} X[k]\varphi_l[2n-k], \tag{2}$$

$$dD = \sum_{k=-\infty}^{\infty} X[k]\varphi_h[2n-k], \tag{3}$$

where X is the original signal, φ represents the filter, and the subscribe l and h represent the low-pass filter and high-pass filter, respectively.

Figure 2(a) demonstrates the process of Mallat algorithm for two-level decomposition. The original time sequence data X is put through both low-pass filter and high-pass filter, and the outputs are $dA1$ and $dD1$ of the first layer, respectively. Then, the obtained $dA1$ is passed through two

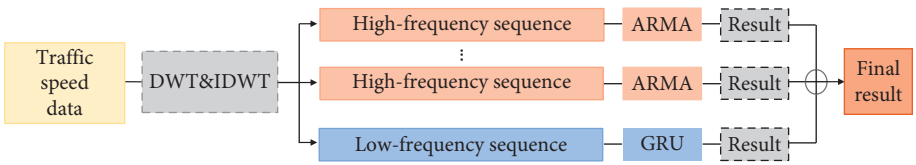


Figure 1: The framework of the proposed model.

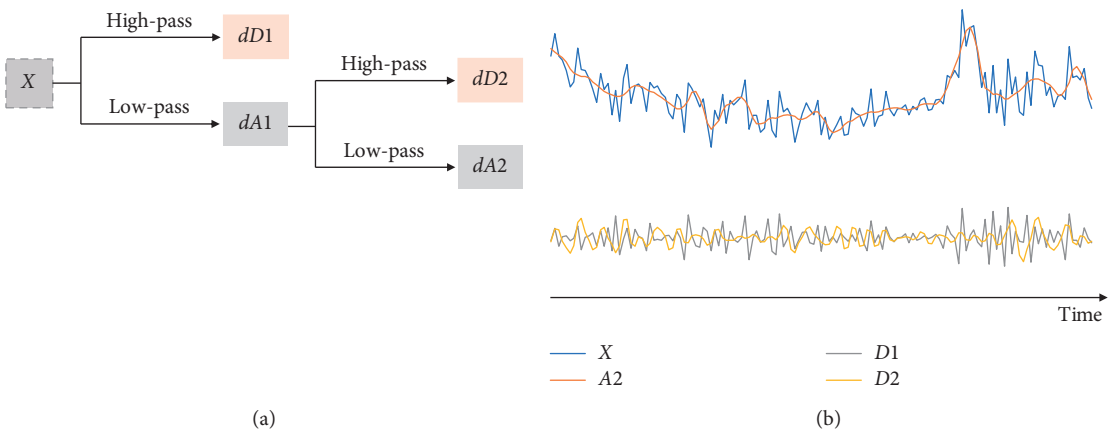


Figure 2: Discrete wavelet transform. (a) Decomposition process. (b) Sequence reconstruction.

filters again to obtain two coefficients $dA2$ and $dD2$. After decomposition, time series components with different frequencies can be obtained, but the lengths of the components after decomposition are not equal. The sequence length is reduced by half after the decomposition. Therefore, inverse discrete wavelet transform (IDWT) is used to reconstruct the data using approximate coefficients and detail coefficients to obtain sequences that are equal to the length of the original sequence but with different frequencies. As shown in Figure 2(b), the approximate coefficient $dA2$ is used to reconstruct the low-frequency component to form a sequence $A2$; the detail coefficients $dD1$ and $Dd2$ are used to reconstruct the high-frequency component to obtain the sequences $D1$ and $D2$.

3.3. Gated Recurrent Unit. RNN has a wide range of applications in the field of time series analysis. It can implement a mechanism similar to the human brain and maintain a certain memory of the processed information. However, traditional RNN models are prone to vanishing gradients and gradient explosions during training [39]. A variation of RNN called LSTM is proposed to solve the problem effectively. The cells of hidden layers for LSTM have a special structure compared with traditional neuron nodes, which is the key to the long-term dependence of LSTM learning time series. Figure 3(a) shows the cell structure of the hidden layer of LSTM. Information inflows, outflows, and previous status updates can be achieved by adding input gates, output gates, and forget gates to this cell structure. The forget gate is responsible for determining how much of the previous cell state is retained in the current cell state, the input gate is responsible for determining how many inputs are retained in the current cell state, and the output gate is responsible for determining the output of the current cell state.

Gated Recurrent Unit (GRU) is a variation of LSTM networks. It inherits the advantages of RNN model: it automatically learns features and effectively models long-term-dependent information. It has been applied to short-term traffic prediction successfully [11, 12]. Figure 3(b) shows the cell structure of the hidden layer of GRU network, and it is more simply compared with LSTM, obviously. Intuitively, the input and forget gates in LSTM were integrated as a reset gate in GRU [9], which determines how to combine the new input information with the information from the previous time. Another gate in GRU is called update gate; it determines how much of the information from the previous time can be saved to the current time. Therefore, GRU is one gate less compared to LSTM. In addition, the cell state and hidden state in LSTM have been integrated as one hidden state in GRU. These changes make the GRU network have fewer parameters and faster training speed and require less data to generalize the model effectively [40]. The calculation formula of GRU is as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]), \tag{4}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]), \tag{5}$$

$$\overline{h}_t = \tanh(W \cdot [r * h_{t-1}, x_t]), \tag{6}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \overline{h}_t. \tag{7}$$

Formulae (4) and (5) show how the update gate z_t and reset gate r_t are calculated in GRU neurons. W_z denotes the weight of z_t, W_r denotes the weight of r_t, and σ denotes the

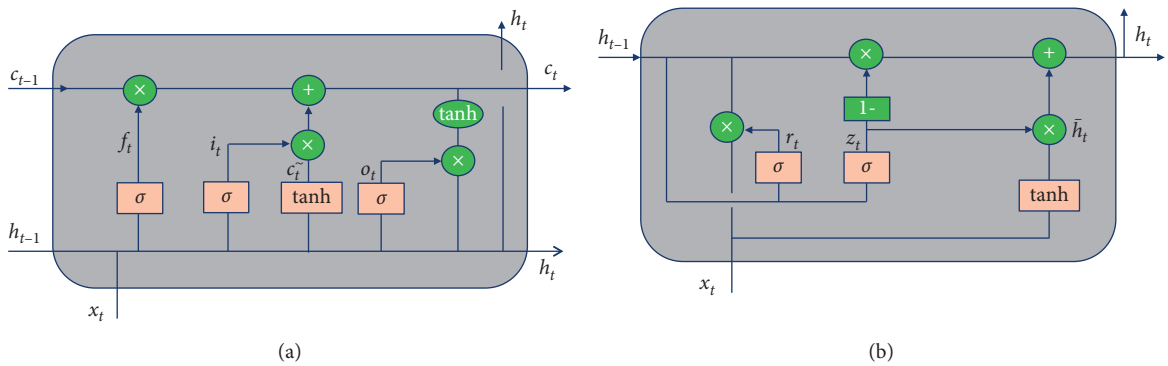


Figure 3: The structure of LSTM and GRU. (a) LSTM. (b) GRU. x_t is input. h_t and h_{t-1} are the output of the current cell and previous cell. In (a), i_t, f_t, and o_t are the input, forget, and output gates, respectively. c_{t-1} and c_t denote the memory cell and the new memory cell content. In (b), r_t and z_t are the reset and update gates, respectively. $\bar{h}_t$ is the pending output.

sigmoid function. The innermost term $[h_{t-1}, x_t]$ represents the sum of vectors h_{t-1} and x_t. A larger value of z_t indicates that more information has been maintained by the current cell while the less for the previous cell. r_t suggests that when the value of the equation is equal to 0, the information from the previous cell is discarded.

Formulae (6) and (7) show the calculation of the pending output value $\bar{h}_t$ and final output value h_t of the GRU neural network. h_{t-1} represents the output from previous cell, W denotes the weight of the z_t, and tanh denotes the hyperbolic tangent function. $\bar{h}_t$ is obtained by multiplying h_{t-1} of the previous cell by r_t, plus x_t, multiplying by the W, and using the hyperbolic tangent function. h_t is the sum of two vectors. One is obtained by multiplying $1-z_t$ by h_{t-1} and another one is obtained by multiplying zt by $\bar{h}_t$.

3.4. Autoregression Moving Average Model. Autoregressive moving average (ARMA) model is the most common type of time series models used for stationary random process analysis [41]. This method does not require strong similarity between the data at the predicted time and the data at the previous time. The method could smooth the predicted values at excessive fluctuations by averaging several measured data. The high-frequency sequences generated by WT have smooth fluctuations. Therefore, this article makes use of ARMA to predict the high-frequency sequences and simulate the random fluctuation of original data caused by the real-time and dynamic nature of traffic. The basic model of the autoregressive moving average model is ARMA (p, q), which consists of two parts, namely, the autoregressive model (AR) and the moving average model (MA). The basic expression is as follows:

$$\tilde{y}_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i}^{high} + \sum_{i=1}^{q} \lambda_i \varepsilon t - i, \tag{8}$$

where c is a constant, ε_t denotes the random error of the Gaussian white noise distribution, φ and λ are the parameters of the models AR and MA, and p and q refer to the orders of the models AR and MA. On the left side of the equation, $\tilde{y}_t$ denotes the predicted result of the ARMA model, corresponding to the predicted value of the traffic speed on the high-frequency sequences at time t.

4. Experiments

4.1. Data. The data used to evaluate the performance of the proposed model were the floating car speed data collected in Ningbo. The raw data were uploaded by the GPS equipment on about 4,300 taxis every day from July 1, 2017, to July 30, 2017. The GPS equipment records the running status of the vehicle every fifteen seconds. Each recorded piece of floating car speed data included record time, vehicle speed (instantaneous speed), location (latitude and longitude), and direction of travel. These data-high-frequency floating vehicle data-can reflect detailed vehicle dynamics [42]. In this study, a representative busy area in Ningbo was selected for the study, and the road network was divided into 283 sections based on the existence of intersections (Figure 4). The vector road network data were obtained from Open Street Map.

In order to use the raw data to estimate the average road speed, data cleaning was performed in the first place. The erroneous data with incorrect time and location have been deleted, and the abnormal speed values were identified and removed by the interquartile range method [43]. Since the speed of urban sections will be affected by the intersection [42, 44], the data uploaded by GPS when the vehicle is suspended at the intersection temporarily were kept, which makes the final estimated results more realistic. Then, the Feature Manipulate Engine (FME) platform is used to estimate the speed of the road sections. FME is a set of solution customization software for spatial and nonspatial data analysis, processing, and conversion. Through this platform, a geometric map matching algorithm [45] is used to match the cleaned spatial data with direction attributes to the road network. Meanwhile, an algorithm for estimating the average vehicle speed at the interval of ten minutes is designed. In the algorithm, the average value of consecutive track points of the same vehicle on the same road section within the same ten minutes represents the average speed of a single vehicle (SV). And the final average speed of a road section (RV) is the mean of the all SV. After all these, the missing values were imputed by the linear interpolation [43]. Finally,

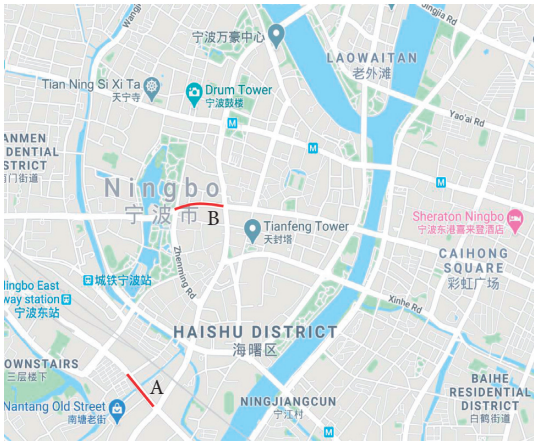


Figure 4: Road networks of the target area.

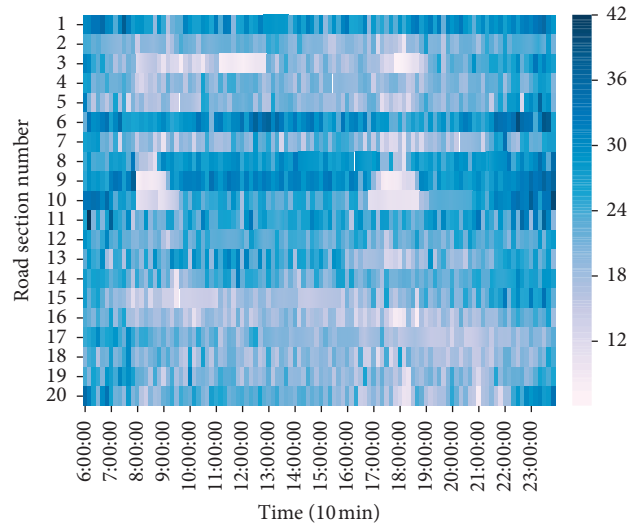


Figure 5: Speed estimation for some road sections of the target area.

148 speed values are obtained every day and we choose data from 6 a.m. to 24 p.m. (108 time steps in 1 day) as experimental data. A time-space traffic diagram, in which x-axis is time, y-axis is space, and the color inside represents speed [46], is used to demonstrate the processed data, as shown in Figure 5.

Next, road section B, which is adjacent to the hospital, and the main channel section A of the city are the main research objects, as shown in Figure 4. The calculated results of the two road sections are shown in Figure 6. Road section B has a low overall speed during the daytime, and the pattern is not obvious. However, road section A has a more obvious morning and evening peak trend. According to the demand of the prediction model and the abovementioned data processing steps, the average speed of the road section in every driving direction from 06:00 to 24:00 every 10 min was obtained. After that, a time series of road section speeds with 108 data per day and a total of 3240 data over 30 days were generated. The road section speed data with an effective length of 30 days were divided into two parts according to a ratio of 8 : 2. The data of the first 24 days were used to train the model, and the data of the next 6 days were used to test the model.

4.2. Accuracy Indicators and Experimental Setup. Two measurements are used as performance indicators for the accuracy of the proposed model of short-term prediction. They are Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE), as shown in Formulae (8) and (9). MAPE is the relative error of the prediction, and RMSE provide the deviation of the predicted value from the actual value. These measurements help us to better understand the prediction results:

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|\tilde{x}_i - x_i\right|}{x_i}, \tag{9}$$

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{n}\frac{\left(\tilde{x}_i - x_i\right)^2}{n}}, \tag{10}$$

where $\tilde{x}$ is the predicted value at time i, x is the actual value at time i, and n is the number of predicted values.

According to the characteristics of wavelet decomposition, DB4 is used as the mother wavelet of DWT to decompose and reconstruct the original data in two stages. Two high-frequency sequences and one low-frequency sequence can be obtained for training and prediction. For low-frequency sequences, the minimum-maximum normalization method is used to scale the input data to the [−1, 1] range before training the model. The predicted output values by the model will be readjusted to normal values. After several experiments, a GRU network with two hidden layers and each hidden layer with 256 units is used to predict low-frequency sequences. All neural network approaches were implemented using Tensorflow.

4.3. Results and Analysis. In this section, a speed data set for urban road sections is used to evaluate the W-GRU-ARMA model. The validity of the model is verified by predicting the speed value of the road section in the next 10 minutes.

First of all, we applied the model to the experimental data of two road sections (A and B) with different traffic patterns. In this prediction experiment, it is important to choose the appropriate time steps for input. The best input is determined after performing prediction experiments on different inputs of time steps from 1 to 8 in the GRU network. Experimental results are shown in Figure 7.

Figure 7 shows that when the input time step increases from 1, the prediction errors decrease rapidly and the best prediction results on both A and B were obtained when the input time step is 2. The RMSE and MAPE values of prediction for road section A are 1.585 and 6.014, respectively, and those of road section B are 1.361 and 5.459, respectively. On the other hand, as the number of input step size continues to increase, the prediction errors do not decrease significantly. This means that the traffic status at a certain time has a strong correlation with the historical time closer

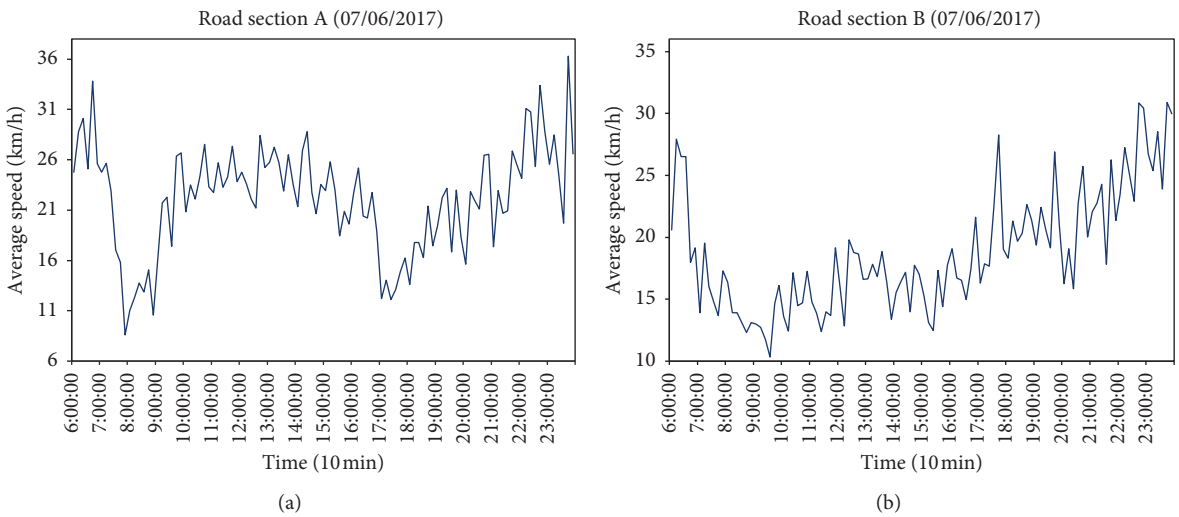


Figure 6: The traffic patterns. (a) Road section A. (b) Road section B.

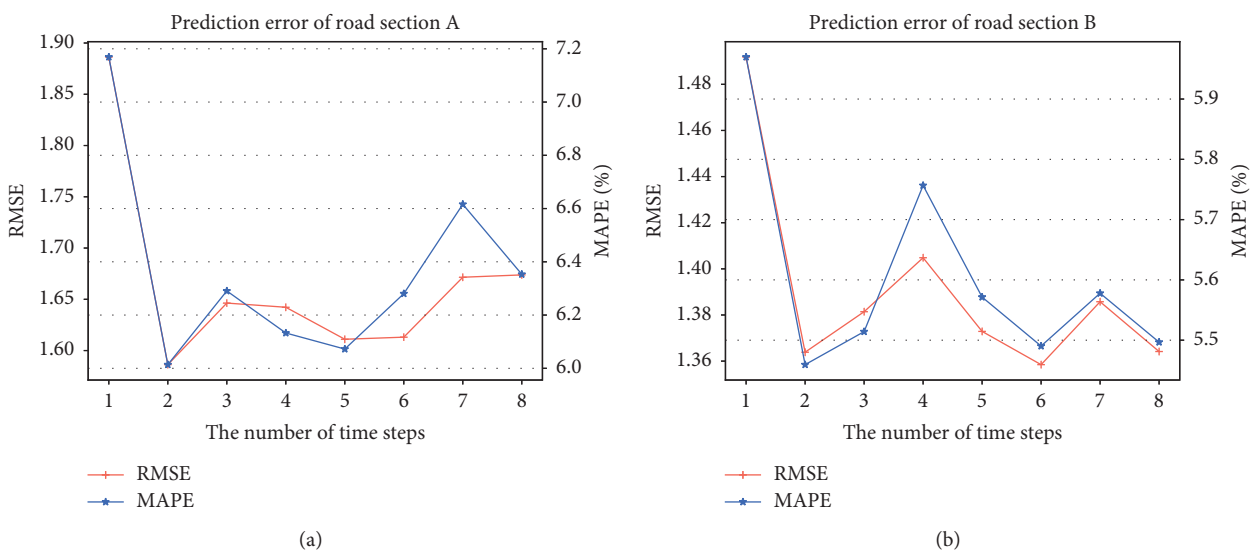


Figure 7: Prediction error of input at different time steps. (a) Road section A. (b) Road section B.

to it. In addition, the prediction errors of the two research sections have similar changes with the increase of the input time step, which may indicate that the traffic data generated by different road sections have similar short-term dependencies. Therefore, the time step of the input was set to 2 in subsequent experiments.

In this prediction experiment, a total of six days of traffic speeds were predicted, including weekdays and weekends. The actual traffic speed, the predicted traffic speed, and the associated residual for road sections A and B on July 26 (Wednesday) and July 30 (Sunday) are shown in Figure 8 to illustrate the prediction results of proposed model.

Figure 8 show that the actual value and the predicted value are in good agreement. The traffic pattern of road section A is more complicated on weekend, which makes the prediction difficult. The proposed model, therefore, has slightly better prediction performance during the weekdays than at the weekends. On the other hand, the position of the standard red rectangles also reflects that the proposed model can better catch sudden changes of traffic speed. In addition, the prediction result drawn in Figures 8(c) and 8(d) reflects that the model performs better on road section B, possibly because similar traffic pattern B presents every day.

In order to find more details from the prediction results, the prediction errors of the two road sections per hour within 6 days are calculated and shown by box diagrams in Figure 9.

As shown in Figure 9, the prediction error of a road section at different times of the day is inequable. No matter for road section A or B, the prediction performance is more stable at the nonpeak hours, and there is a certain fluctuation of the performance at the morning and evening travel peaks,

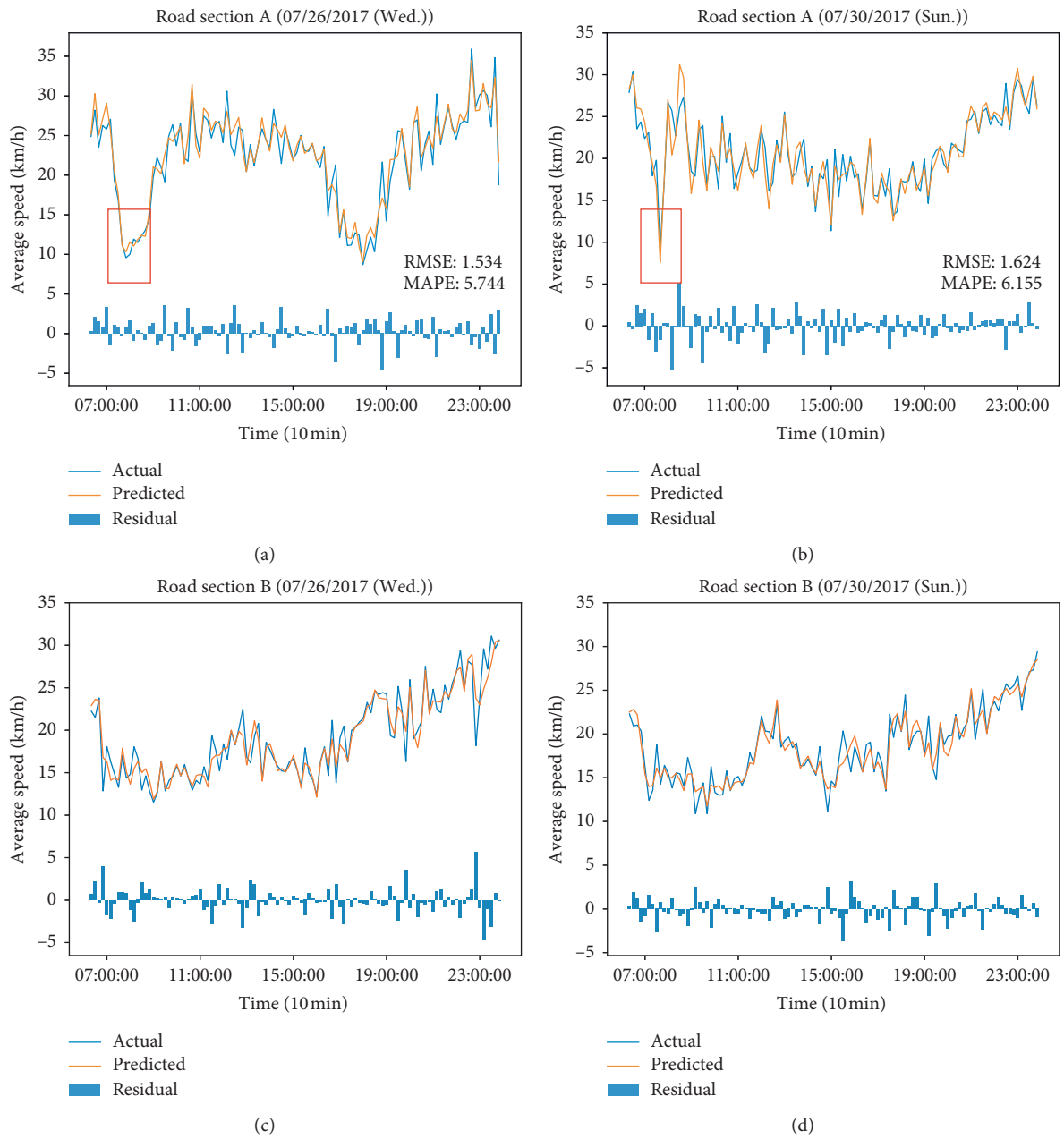


Figure 8: The predicted result of road sections A and B. (a) Road section A on 07/26/2017. (b) Road section A on 07/30/2017. (c) Road section B on 07/26/2017. (d) Road section B on 07/30/2017.

especially on road section A. For road section A, the prediction performance is better at noon and evening. Furthermore, the regularity of the prediction result for road section B in different time periods is not significant, because road section B is relatively busy all the time. But it is worth mentioning that the prediction error in different periods is always within a smaller range.

To verify the effectiveness of the proposed model, the prediction performance is compared with the GRU model, LSTM model, SAEs model, and ARIMA model. Tables1 and 2 demonstrate the prediction performance of different models.

From Tables 1 and 2, it can be seen that the prediction performance of the proposed model is better than ARIMA, GRU, SAEs, and LSTM models on the two experimental road sections, especially compared with ARIMA. This is owing to the ARIMA model which assumes that the traffic condition is a stationary process while this is not always true in reality. In general, the prediction error of each model on road section B is lower than that on road section A. In the prediction for road section B, the performance of LSTM is better than GRU. Based on this finding, this experiment was expanded on 12 road sections. The experimental results of the following four models are compared:

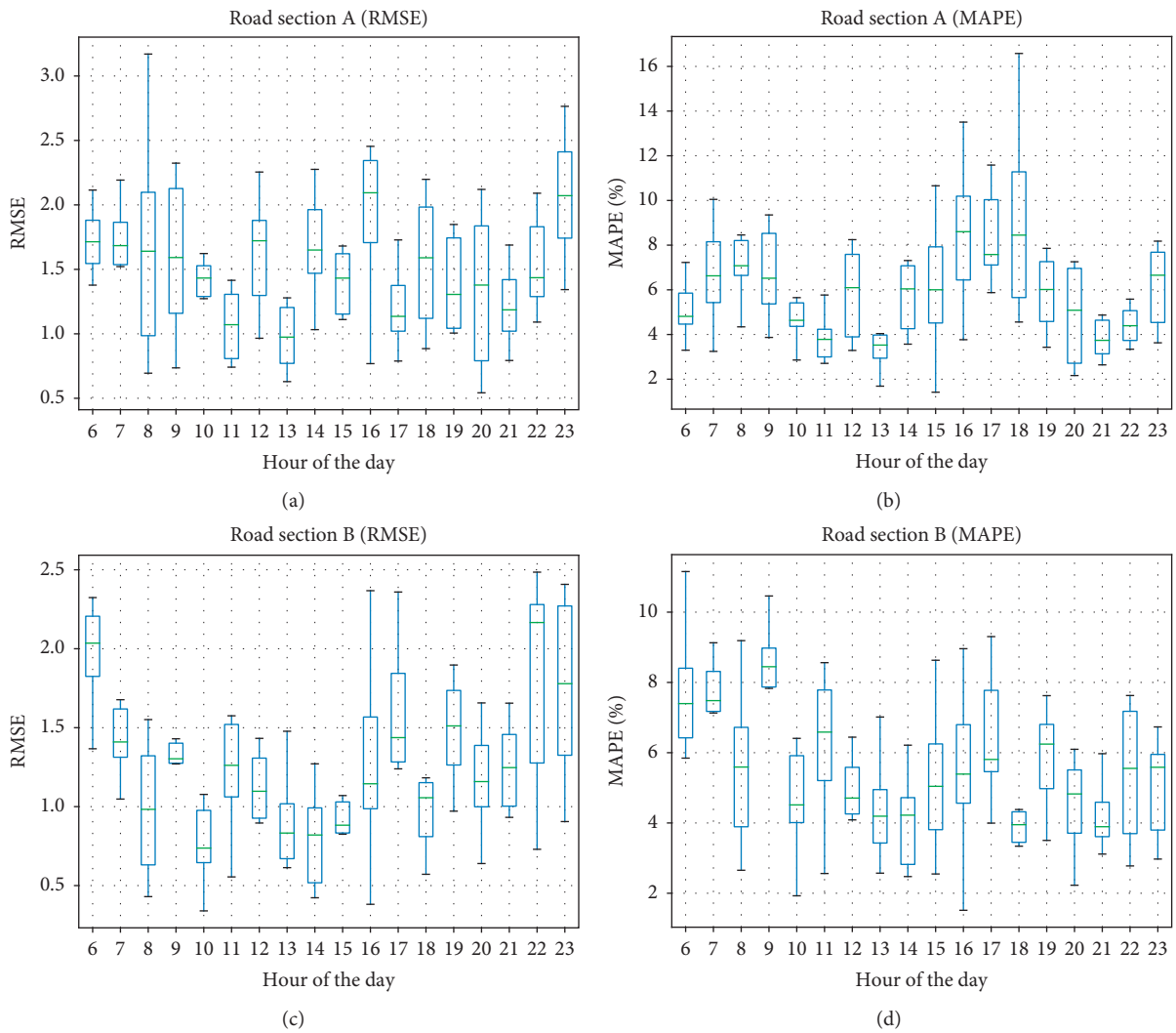


Figure 9: The error distribution per hour. (a) The RMSE of road section A, (b) the MAPE of road section A, (c) the RMSE of road section B, and (d) the MAPE of road section B.

Table 1: Performance comparison of different models on road section A.

Model	RMSE	MAPE (%)
ARIMA	2.104	9.593
SAEs	1.874	7.721
LSTM	1.946	8.132
GRU	1.860	7.656
W-GRU-ARMA	**1.585**	**6.014**

W + GRU + ARMA, W + LSTM + ARMA, GRU, and LSTM. The prediction results are presented in Figure 10.

As can be seen in Figure 10, W + GRU + ARMA achieves better results than W + LSTM + ARMA when the prediction results of the ARMA model are the same. This means that GRU networks, which are simpler in structure than LSTM, have the potential to predict the low-frequency sequence. Looking at it another way, the prediction effect of W + LSTM + ARMA model is not as good as that of the pure LSTM on some road sections. This is because pure LSTM can better capture the sudden changes in traffic data [26]. When LSTM is used instead of GRU in our proposed model, it is possible that the smoothness of the data makes LSTM lose its advantage in some cases.

To verify the robustness of the proposed model, W-GRU-ARMA prediction was applied to the experimental data of 81 road sections in the study area. The prediction performance of different models is shown in Figure 11.

Prediction results shown in Figure 11 have proven that the proposed model is robust. The prediction error of the

Table 2: Performance comparison of different models on road section B.

Model	RMSE	MAPE (%)
ARIMA	2.220	8.799
SAEs	1.376	6.077
LSTM	1.367	6.398
GRU	1.582	7.636
W-GRU-ARMA	**1.361**	**5.459**

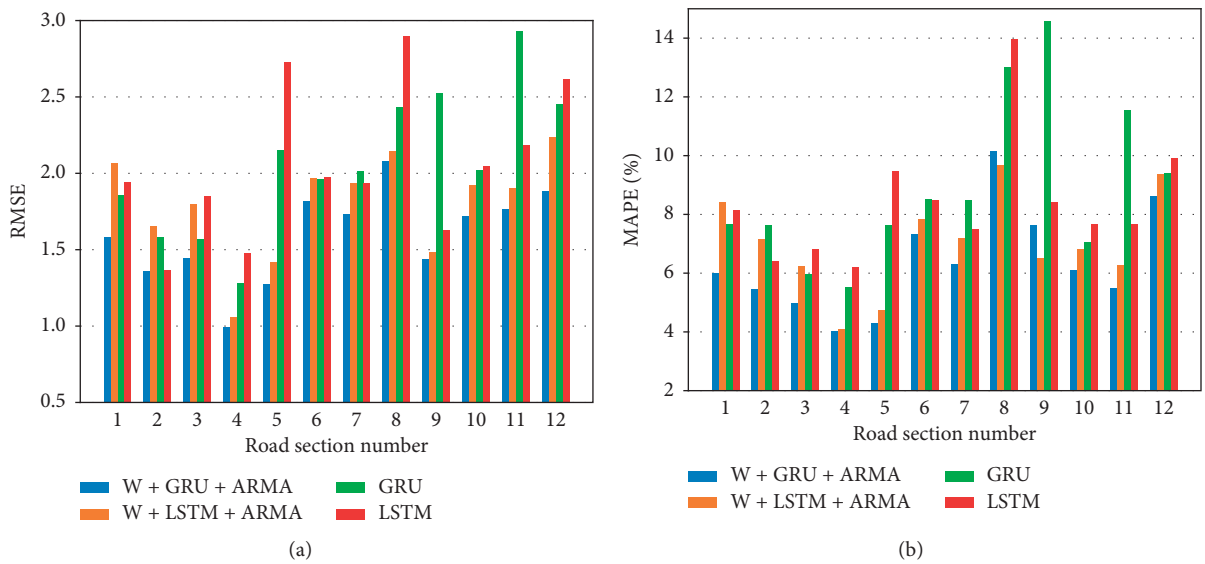


Figure 10: Prediction performance of four models on 12 road sections. (a) RMSE and (b) MAPE.

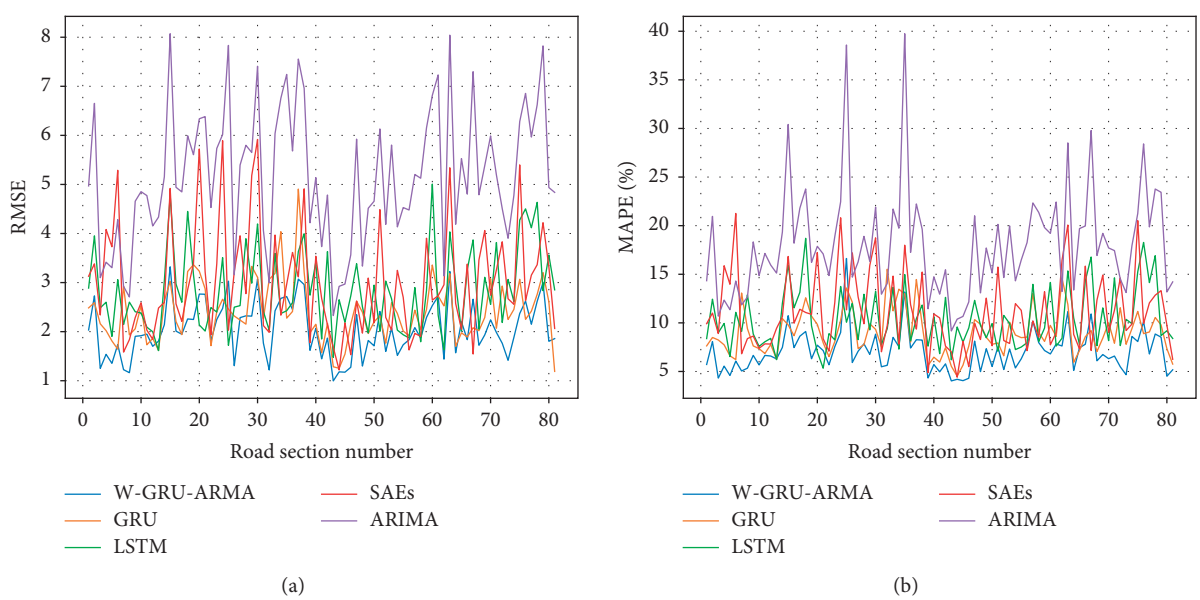


Figure 11: Continued.

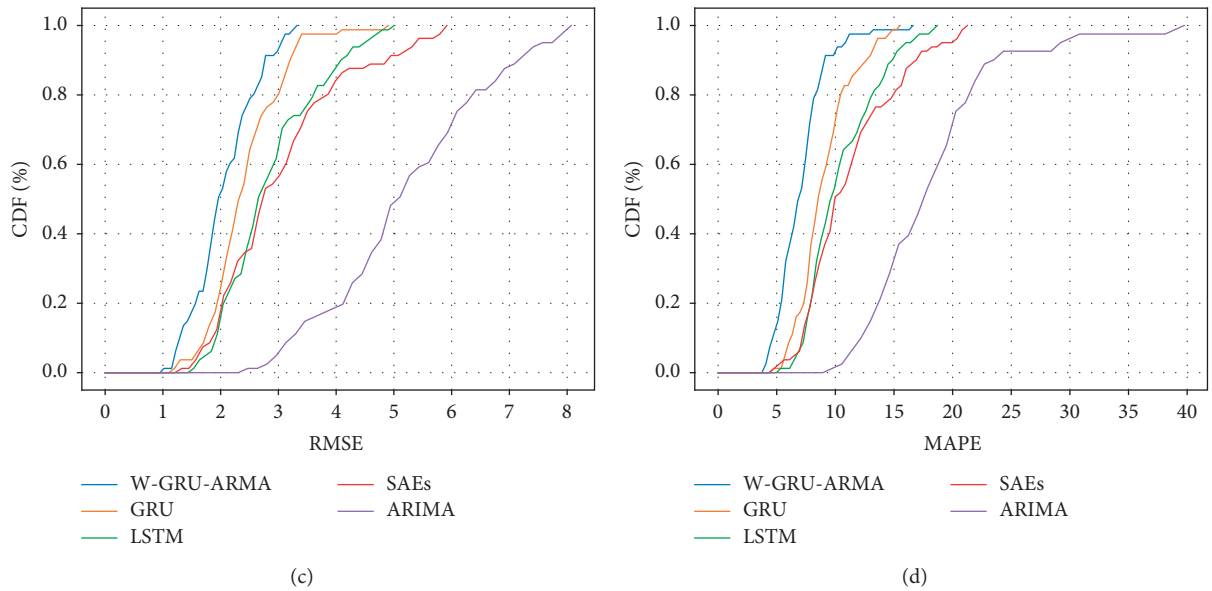


Figure 11: Prediction performances comparison of different models. (a) The RMSE of 81 road sections, (b) the MAPE of 81 road sections, (c) the Cumulative Distribution Function of RMSE, and (d) the Cumulative Distribution Function of MAPE.

proposed model always fluctuates within a lower range, while the performance is better than other models on most road sections. On more than 80% road sections, according to Figure 11(c), the RMSE of prediction is below 2.5. The MAPE, as shown in Figure 11(d), is below 10% on more than 90% of the road sections. These results verify the effectiveness and superiority of our model for short-term traffic prediction on road sections.

5. Conclusions

This paper proposed a new combination model for short-term vehicle speed prediction based on the spatiotemporal correlation of traffic data and unstable random fluctuations. The model consists of three components, namely, WT, GRU network, and ARMA model. For one thing, the historical traffic speed data are decomposed into subsequence from high to low in terms of frequencies by WT. Furthermore, there are multiple low-frequency sequences that have stable trends and spatiotemporal relationships between one another. A GRU model is constructed so that it takes in these sequences and then predicts the speed value of next moment. At the same time, the ARMA model is used to predict high-frequency sequences with unstable randomness. This enables our proposed model to simultaneously fit the steady trend and randomness of traffic speed data. The prediction experiment using the real traffic speed data generated by floating vehicle was carried out. Experimental results showed the advantage of the proposed model compared with the previous models in terms of two performance measures: MAPE and RMSE.

We also obtained the following key content from the experimental results. When GRU networks and LSTM networks are used to predict low-frequency sequences, GRU networks perform better, which means that GRU networks have great potential for smooth sequence prediction. In addition, although the traffic speed data has a strong time dependence, the prediction performance of the model does not increase with the input time step. It makes sense to choose the best input before conducting a prediction experiment. It is also found that the prediction error of the proposed model in different periods is slightly different, implying that effective analysis of traffic data at different time periods and the establishment of corresponding models may improve the accuracy of traffic predictions. Therefore, this will be the main content of our further research.

Data Availability

The taxi trajectory data used to support this study were provided by the Ningbo public transportation passenger transport administration and were made nonpublic.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge the Fundamental Research Funds for the Central Universities (no. 300102230501).

Supplementary Materials

Supplementary files are the original data of the experimental part of the paper. These data are used to train and test the model to verify its validity and accuracy. File “road_A.xlsx” is the training data of road section A, and file “road_A_test.xlsx” is the test data of road section A. File

"road_B.xlsx" is the training data of road section B, and file "road_B_test.xlsx" is the test data of road section B. Each file contains four valid fields: "time," "up," "mid," and "down," where "up" represents the speed data of the upstream section, "mid" represents the speed data of the study, and "down" represents the speed data of the downstream section. (*Supplementary Materials*)

References

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: where we are and where we'e going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.

[2] X. Zhang, G. Xiong, L. Xiao et al., "A design of intelligent route guidance system based on shortest path algorithm," in *Proceedings of the 2015 IEEE International Conference on Service Operations And Logistics, And Informatics (SOLI)*, pp. 12–17, IEEE, Hammamet, Tunisia, November 2015.

[3] U. Ryu, J. Wang, T. Kim, S. Kwak, and J. U, "Construction of traffic state vector using mutual information for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 96, pp. 55–71, 2018.

[4] J. Tang, L. Li, Z. Hu, and F. Liu, "Short-term traffic flow prediction considering spatio-temporal correlation: a hybrid model combing type-2 fuzzy C-means and artificial neural network," *IEEE Access*, vol. 7, 2019.

[5] A. Garcia-Ortiz, S. M. Amin, and J. R. Wootton, "Intelligent transportation systems—enabling technologies," *Mathematical and Computer Modelling*, vol. 22, no. 4–7, pp. 11–81, 1995.

[6] X. Ma, H. Yu, Y. Wang et al., "Large-scale transportation network congestion evolution prediction using deep learning theory," *PloS One*, vol. 10, no. 3, 2015.

[7] B. Yu, X. Song, F. Guan et al., "k-Nearest neighbor model for multiple-time-step prediction of short-term traffic condition," *Journal of Transportation Engineering*, vol. 142, no. 6, Article ID 4016018, 2016.

[8] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387–399, 2011.

[9] K. Cho, B. Van Merrinboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[10] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the International Conference on Machine Learning*, International Machine Learning Society (IMLS), Lille, France, pp. 2342–2350, July 2015.

[11] G. Dai, C. Ma, and X. Xu, "Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU," *IEEE Access*, vol. 7, pp. 143025–143035, 2019.

[12] Y. Gu, W. Lu, L. Qin, M. Li, and Z. Shao, "Short-term prediction of lane-level traffic speeds: a fusion deep learning model," *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 1–16, 2019.

[13] N. Zhang, X. Guan, J. Cao, X. Wang, and H. Wu, "Wavelet-HST: a wavelet-based higher-order spatio-temporal framework for urban traffic speed prediction," *IEEE Access*, vol. 7, pp. 118446–118458, 2019.

[14] Y. Gao and F. Chen, "Wavelet analysis-based npr prediction of short-term traffic flow," *Journal of University of Science & Technology of China*, vol. 38, no. 12, pp. 1427–1431, 2008.

[15] L. C. Ma, W. L. Xu, and D. X. Liu, "Prediction model of traffic flow along typical roads in city urban district based on wavelet transform," *Control and Decision*, vol. 26, no. 5, pp. 789–793, 2011.

[16] C. Yanchong, H. Darong, and Z. Ling, "A short-term traffic flow prediction method based on wavelet analysis and neural network," in *Proceedings of the 2016 Chinese Control and Decision Conference (CCDC)*, pp. 7030–1034, IEEE, Yinchuan, China, May 2016.

[17] K. I. Wong and Y. C. Hsieh, "Short-term traffic flow forecasting for urban roads using space-time ARIMA," in *Proceedings of the Transportation and Urban Sustainability*, Hong Kong Society for Transportation Studies, Hong Kong China, pp. 583-584, December 2010.

[18] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *European Transport Research Review*, vol. 7, no. 3, p. 21, 2015.

[19] L. L. Ojeda, A. Y. Kibangou, and C. C. De Wit, "Adaptive Kalman filtering for multi-step ahead traffic flow prediction," in *Proceedings of the 2013 American Control Conference*, pp. 4724–4729, IEEE, Washington, DC, USA, June 2013.

[20] S. V. Kumar, "Traffic flow prediction using kalman filtering technique," *Procedia Engineering*, vol. 187, pp. 582–587, 2017.

[21] M. T. Asif, J. Dauwels, C. Y. Goh et al., "Spatiotemporal patterns in large-scale traffic speed prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 794–804, 2013.

[22] B. Yao, C. Chen, Q. Cao et al., "Short-term traffic speed prediction for an urban corridor," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 2, pp. 154–169, 2017.

[23] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 21–34, 2016.

[24] D. Xia, B. Wang, H. Li, Y. Li, and Z. Zhang, "A distributed spatial-temporal weighted model on MapReduce for short-term traffic flow forecasting," *Neurocomputing*, vol. 179, pp. 246–263, 2016.

[25] C. Gang, W. Shouhui, and X. Xiaobo, "Review of spatio-temporal models for short-term traffic forecasting," in *Proceedings of the 2016 IEEE international conference on intelligent transportation engineering (ICITE)*, pp. 8–12, IEEE, Singapore, August 2016.

[26] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, Neural Information Processing Systems Foundation, Inc. (NIPS), December 2012, pp. 1097–1105, Lake Tahoe, NV, USA.

[28] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[29] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, AAAI, Stockholm, Sweden, July 2018.

[30] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: data-driven traffic forecasting," in

Proceedings of the International Conference on Learning Representations (ICLR'18), International Conference on Learning Representations, Vancouver, BC, Canada, May 2018.

[31] Y. Lv, Y. Duan, W. Kang et al., "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[32] Y. Jia, J. Wu, and Y. Du, "Traffic speed prediction using deep learning method," in *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1217–1222, Rio de Janeiro, Brazil, IEEE, November 2016.

[33] G. Fusco, C. Colombaroni, and N. Isaenko, "Short-term speed predictions exploiting big data on large urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 73, pp. 183–201, 2016.

[34] F. Lin, Y. Xu, Y. Yang et al., "A spatial-temporal hybrid model for short-term traffic prediction," *Mathematical Problems in Engineering*, vol. 2019, Article ID 4858546, 12 pages, 2019.

[35] Z. Duan, Y. Yang, K. Zhang, Y. Ni, and S. Bajgain, "Improved deep hybrid networks for urban traffic flow prediction using trajectory data," *IEEE Access*, vol. 6, pp. 31820–31827, 2018.

[36] L. Ouyang, F. Zhu, G. Xiong et al., "Short-term traffic flow forecasting based on wavelet transform and neural network," in *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, Yokohama, Japan, October 2017.

[37] C. Parameswariah and M. Cox, "Frequency characteristics of wavelets," *IEEE Transactions on Power Delivery*, vol. 17, no. 3, pp. 800–804, 2002.

[38] A. Jensen and A. la Cour-Harbo, *Ripples in Mathematics: The Discrete Wavelet Transform*, Springer Science & Business Media, Berlin, Germany, 2001.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[40] L. Kuan, Z. Yan, W. Xin et al., "Short-term electricity load forecasting method based on multilayered self-normalizing GRU network," in *Proceedings of the 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1–5, IEEE, Beijing, China, November 2017.

[41] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic timeseries data by using box-jenkins techniques," *Transportation Research Record*, vol. 773, no. 722, pp. 1–9, 1979.

[42] Z. He, G. Qi, L. Lu, and Y. Chen, "Network-wide identification of turn-level intersection congestion using only low-frequency probe vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 108, pp. 320–339, 2019.

[43] H. Yu, N. Ji, Y. Ren, and C. Yang, "A special event-based K-nearest neighbor model for short-term traffic state prediction," *IEEE Access*, vol. 7, pp. 81717–81729, 2019.

[44] Y. Yue, H. X. Zou, and Q. Q. Li, "Urban road travel speed estimation based on low sampling floating car data," in *Proceedings of the ICCTP 2009: Critical Issues In Transportation Systems Planning, Development, and Management*, pp. 1–7, American Society of Civil Engineers, Reston, VA, USA, July 2009.

[45] J. S. Greenfeld, "Matching GPS observations to locations on a digital map," in *Proceedings of the 81th Annual Meeting of the Transportation Research Board*, Transportation Research Board (TRB), Washington, DC, USA, pp. 164–173, April 2002.

[46] Z. He, Y. Lv, L. Lu, and W. Guan, "Constructing spatio-temporal speed contour diagrams: using rectangular or non-rectangular parallelogram cells?" *Transportmetrica B: Transport Dynamics*, vol. 7, no. 1, pp. 44–60, 2019.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 7317480, 12 pages
https://doi.org/10.1155/2020/7317480

Research Article

Does Patent Help to Build Investment Portfolio of China A-Shares under China-US Trade Conflict?

Yu-Jing Chiu,[1] Kuang-Chin Chen,[1] and Hui-Chung Che[2]

[1]*Department of Business Administration, Chung Yuan Christian University, Taoyuan, Taiwan*
[2]*Shenzhen TekGlory Intellectual Property Data Technologies, Ltd., Shenzhen, China*

Correspondence should be addressed to Hui-Chung Che; drcharlie918@yeah.net

Received 25 March 2020; Accepted 17 April 2020; Published 13 May 2020

Guest Editor: Chi-Hua Chen

Patent, a legal representation of innovation achievement, is strongly meaningful for almost every country's economic growth and technology development. China, the world's no. 2 stock market, is the world's largest patent application country. In this study, we observed 2,197 China-listed companies of Renminbi (RMB) common stocks (A-shares) from 2016 to 2018. The relationship among 570 valid patent indicators and financial indicators of the stock price, Return-on-Assets (ROA), and Return-on-Equity (ROE) was examined. We constructed patent leading indicators, patent kernel indicators, and patent prediction equations for predicting the stock price, ROA, and ROE using the Granger causality test and the dynamic time series forecast model. The stock performance of investment portfolios based on patent prediction equations was thoroughly discussed. We found that investment portfolios constructed by the higher predictive ROA, the higher predictive ROE, and the higher predictive stock price growth rate have better performance than the A-shares' average. The underlying concept behind this study is that despite the overall economic environment fluctuation and the China-US trade conflict, the patent-based algorithm proposed was proved to be useful to discover good investment portfolios.

1. Introduction

Global economic growth seems to be losing momentum in 2019. Productivity growth has recorded a decline, trade wars continue, and economic uncertainty remains high. Despite the sluggish market sentiment, innovation is in full swing around the world. Whether developed economies or developing economies, innovation activities (which can be measured by R&D and patents) are booming, and innovation spending has been increasing.

When the GDP of China surpassed that of Japan, China became the second-largest economy in the world since 2010. The value of China stock market (there are two stock exchanges in China: one in Shanghai and the other in Shenzhen. The A-share stocks comprise four stock boards: Shanghai main board, Shenzhen main board, GE board, and SME board. Stock codes of the Shanghai main board start with 600, 601, and 603; most stocks are of state-owned listed companies and big companies. Stock codes of the Shenzhen main board start with 000 and 001; most stocks are of state-owned listed companies and big companies. Stock codes of the GE board start with 300; most stocks are of small and medium companies. Stock codes of the SME board start with 002; most stocks are of small & medium companies), comprising more than 3,400 listed companies of RMB common stocks (A-shares), is also ranked as the world's no. 2.

According to the World Intellectual Property Organization, for years, China has been the largest source of patent applications. In 2018, 4.32 million patent applications emanated from China (China patent applications comprise three species of submitted and undisclosed applications: invention applications, utility model applications, and design application). By the end of 2018, the cumulative number of patent publications (China patent publications comprise four species of publications: invention publications, invention grants which have been disclosed and have passed substantial examination, utility model grants which have

been disclosed and have passed the initial examination, and design grants which have been disclosed and have passed the initial examination) in China's database reached 22 million. China's patent database is the largest in the world.

The main objective of investing in technological innovation is to have commercial benefits in the future. Deng et al. [1] and Thomas [2] proposed the use of multiregression analysis for modeling the patent indicators and the price-to-book ratio in the US stock market. Though the adjusted R^2 was very poor, they still found a way to distinguish the undervalued and overvalued stocks. For China, with its enormous number of patents and the size of the stock market, is it possible to find a similar relationship between patent indicators and stock performance? Chen et al. [3] modified the abovementioned multiregression model for discussing the data of Shanghai A-shares from 2011 to 2017. The leading and lagging relationship between the patent indicators and the stock price was roughly modeled. The patent leading equation for predicting the stock price by patent leading indicators was proposed. It seems that under the normal macroeconomic environment, the investment portfolio selected based on the higher predictive stock price return rate performed better than the market trend.

Unfortunately, on March 22, 2018, the US government launched a trade war against China through the tariff system. US President Trump officially signed a trade memorandum announcing that the US will impose tariffs on 60 billion US dollars' worth of goods imported from China and restrict Chinese companies' investment, mergers, and acquisitions in the United States. On April 4, 2018, the US government released a list of goods subject to tariffs, which amounted to the imposition of a 25% tariff on approximately US$50 billion worth in goods imported from China. On April 5, 2018, US President Donald Trump instructed the United States Trade Representative to consider imposing US$100 billion in additional tariffs on imports from China. On July 6, 2018, the first batch of US$34 billion worth of Chinese goods entering the US faced the new 25% tariff. The China-US trade conflict not only seriously affects China's exports to the US but also impacts the Chinese stock market. From the beginning to the end of 2018, the CSI 300 Index (CSI 300 Index, code 000300, is composed of 300 large-scale, liquidity, and most representative high-quality stocks selected from the entire list of A-shares and represents the top stocks in China) declined by 25.3% and the Shanghai Composite Index (Shanghai Composite Index, code 000001, comprises all Shanghai A-shares and represents the market trend of the Shanghai main board stocks) declined by 24.6%.

Against the backdrop of the China-US trade conflict, do the patent indicators proposed by Chen et al. [3] hold good in leading the stock price? Do patent leading indicators also work to correctly predict other financial indicators? Do patent-based stocks still perform better than the market trend? This study aims at answering the above questions.

2. Literature Reviews

R&D capability and market structure are important factors for driving a company's growth and maintaining competitive advantages. Branch [4] found that an increase in the number of patents of a company usually resulted in predictive growth in sales and profits. Griliches [5] found a significant relationship between the market value of a company's shares and the number of patents the company holds. Cockburn and Griliches [6] found an interaction between the effectiveness of patents and the market's valuation of a firm's past R&D and patenting performance. Hall et al. [7] found that citations significantly affect market value. Every additional citation per patent boosts the market value of the stock by 3%. Branch and Chichirau [8] found that patent counts and patent citations are both positively related to growth and negatively related to profitability. Crossan and Apaydin [9] found that the company's innovation results are significantly and positively related to its earnings. Pandit et al. [10] examined whether R&D expenditures and patent counts and citations and their interaction are associated with the level and variability of future earnings and operating cash flows. Fabrizi et al. [11] proposed that venture capitalists are fairly good at judging the value of an idea and whether it is marginally costlier to patent low-value ideas than high-value ideas. Hirshleifer et al. [12] found that innovative efficiency—patents or citations scaled by R&D expenditures—is a strong positive predictor of future returns. Caner et al. [13] theorized that diversity in a firm's patent and alliance portfolios sends contrasting flow signals impacting its market value in a nuanced way. Yu and Hong [14] found that the number of patents has more significant explanatory power than R&D expenditures, incorporating the number of patents in explaining stock returns could add value. Mama [15] found that the relationship of a firm's innovative efficiency and future return is robustly U-shaped.

Deng et al. [1] and Thomas [2] proposed that the main objective of technology analyses is to understand how investing in technological innovation can have commercial benefits. They used quantitative patent indicators in modeling the company price-to-book ratio (PB) by multiregression analysis for the US stock market. They proposed that those companies with patent-modeled PB higher than actual PB are undervalued and those companies with patent modeled PB less than actual PB are overvalued. The highly undervalued stocks might have high investment potential.

All the above studies are about the US stock market. The relationship between the China stock market and patents is rarely discussed. For this purpose, Chen et al. [3] modified the abovementioned multiregression model to discuss Shanghai A-shares in China from 2011 to 2017. They attempted to use the Granger causality test, time series model, and patent prediction equation for predicting the stock price by patent leading indicators. They showed that the stock selection strategy by the predictive stock return rate yields a better stock performance than the market trend.

In this study, we applied the model of Chen et al. [3] to all A-shares rather than to Shanghai A-shares only to check the model's effectiveness. We also modified the model by setting ROA and ROE as the predictive targets because these are more relevant indicators of the company's managerial and operating performance than the stock price. Furthermore,

we discussed the stock portfolio performance based on predictive stock price, predictive ROA, and predictive ROE. Finally, we proposed the investment strategies that may be preferred as workable under the impact of the China-US trade conflict.

3. Methodology

3.1. Population and Sample. China has two stock exchanges, one in Shanghai and the other in Shenzhen, where China A-shares are traded and are spread into Shanghai main board, Shenzhen main board, GE board, and SME board, and these stocks are included in this study. Chinese companies listed on the Hong Kong Stock Exchange and overseas are excluded. As of now, the total number of A-shares is more than 3,400 and is still increasing.

An effective sample of this study must meet two conditions:

(1) During the eight quarters from 2016Q4 to 2018Q3, it remained listed

(2) In each of the eight quarters from 2016Q4 to 2018Q3, it had a new patent publication within one year back through the end of each quarter, the patent may be of any of the patent species mentioned in footnote 3 on page 3

For those A-shares of companies whose subsidiaries' revenue is merged with that of the parent company in the annual report, we assume that patents of subsidiaries have corresponding contributions to the parent company, so patents of such subsidiaries are also merged with patents of the patent company for processing patent indicators.

Table 1 shows effective samples and A-share statistics. A total of 2,197 effective stocks were extracted from all 3,467 A-shares. Of these, the most, that is, 776 effective samples (35.3%) were of the Shanghai main board. However, the largest percentage of effective stocks out of the stocks on any board to be picked was 74.8% from the SME board.

3.2. Patent Indicator. Though China is the leading applicant for patents in the world, only 5% of China's patent applications are filed in foreign countries by A-share companies. The number of patent applications filed abroad for US patents, PCT patents, European patents, etc. is negligible. Therefore, we focused only on patent applications filed in China as the patent indicator.

For boosting industrial innovation, the China government has been subsidizing the fee for filing new patent applications. Many companies apply for a large number of patents to get subsidies and give up unimportant patents when the annual fees for those patents are due. Therefore, only valid patents (valid patents include issued, for which the annual fee has been paid, patents with invention grants, utility model grants, and design patents, and unissued invention publications which are under examination) were included in this study. The valid patent indicators applied in this study are identified as PAij, where $i = 1$ to 10-year data collection interval (to avoid confusion, 10 is represented by X hereinafter) and $j = 1$ to 41, 45 to 60, a total of 57 valid patent indicators for each data collection interval, and a total of 570 patent indicators for all data collection intervals, and the definition is shown in Table 2. The valid patent indicators are extracted from China patent raw data which published by the China National Intellectual Property Administration and included data on invention publications, invention grants, utility model grants, design grants, and legal status data thereof.

When subjected to the Kolmogorov–Smirnov test, the original data distribution of valid patent indicators was observed to be seriously skewed. Therefore, all valid patent indicators applied in this study were Cox-Box transformed to reduce the skewness.

3.3. Financial Indicator. There are three financial indicators of A-shares used in this study, including the stock price, ROA, and ROE. The financial indicators are calculated based on the information revealed by the stock exchanges, quarterly reports, half-yearly reports, and annual reports.

For the stock price, the closing price of the last trading day of each quarter is considered.

ROA is the ratio of the net profit after tax to total assets. ROA indicates how well a company utilizes its assets. ROA is the best indicator for comparing similar companies or comparing a company's ROA at one point in time with the ROA at various points in the past. In this study, ROA at each quarterly settlement is considered.

ROE is the ratio of net profit after tax to shareholders' equity. ROE is considered a measure of how effectively management is using a company's net assets to create profits. In this study, ROE at each quarterly settlement is considered.

3.4. Panel Data-Modeling Period. 2016Q4~2018Q3, a total of eight quarters, is considered before and after the commencement of the China-US trade war.

3.5. Patent Leading Indicator and Patent Kernel Indicator. The Granger causality test is applied for finding the patent leading indicator (hereinafter, PLI). The Granger causality test is a statistical test of hypothesis for determining whether a time series variable is useful in forecasting another. It is not for determining a true cause-and-effect relationship but for finding a probabilistic account of causality. It uses empirical data sets to find leading/lagging patterns of correlation.

For finding the PLI, each of the PAij is applied as a one-time series variable, and each of the financial indicators is applied as the other time series variables. When any of PAij satisfies the Granger causality test (F-test, $p^* < 0.05$) under the lag condition, the PLI is obtained. Lag = 1 means that the leading period of the PLI to the financial indicator is one quarter; Lag = 2 means two quarters; Lag = 3 means three quarters; and Lag = 4 means four quarters. When a PLI satisfies the Granger causality test (the F-test, $p^* < 0.05$) for all lags 1 to 4, it is defined as the patent kernel indicator (hereinafter, PKI).

Table 1: China A-share statistics of 2018Q3.

	Shanghai main board	Shenzhen main board	GE board	SME board	Total
A-shares	1,389	465	710	903	3,467
Effective samples (percentage of total)	776 (35.3%)	258 (11.7%)	488 (22.2%)	675 (30.7%)	2,197
Sampling rate for effective samples percentage of top row	55.9%	55.5%	68.7%	74.8%	63.4%

Table 2: Valid patent indicator PAij.

Valid patent indicator ($i = 1 \sim 10$)	Indicator definition
PAi01	Counts of valid invention publications for i years
PAi02	Counts of valid utility model grants valid for i years
PAi03	Counts of valid design grants for i years
PAi04	Counts of valid invention grants for i years
PAi05	Average of the patent examination duration of valid invention publications for i years
PAi06	Total International Patent Classification (hereinafter, IPC) counts of valid invention publications for i years
PAi07	Total IPC counts of valid utility model grants for i years
PAi08	Total IPC counts of valid invention grants for i years
PAi09	Average IPC counts of valid invention publications for i years
PAi10	Average IPC counts of valid utility model grants for i years
PAi11	Average IPC counts of valid invention grants for i years
PAi12	Total specification words of valid invention publications for i years
PAi13	Total specification words of valid utility model grants for i years
PAi14	Total specification words of valid invention grants for i years
PAi15	Average specification words of valid invention publications for i years
PAi16	Average specification words of valid utility model grants for i years
PAi17	Average specification words of valid invention grants for i years
PAi18	Total claim counts of valid invention publications for i years
PAi19	Total claim counts of valid utility model grants for i years
PAi20	Total claim counts of valid invention grants for i years
PAi21	Average claim counts of valid invention publications for i years
PAi22	Average claim counts of valid utility model grants for i years
PAi23	Average claim counts of valid invention grants for i years
PAi24	Total independent claim counts of valid invention publications for i years
PAi25	Total independent claim counts of valid utility model grants for i years
PAi26	Total independent claim counts of valid invention grants for i years
PAi27	Average independent claim counts of valid invention publications for i years
PAi28	Average independent claim counts of valid utility model grants for i years
PAi29	Average independent claim counts of valid invention grants for i years
PAi30	Total drawing counts of valid invention publications for i years
PAi31	Total drawing counts of valid utility model grants for i years
PAi32	Total drawing counts of valid invention grants for i years
PAi33	Average drawing counts of valid invention publications for i years
PAi34	Average drawing counts of valid utility model grants for i years
PAi35	Average drawing counts of valid invention grants for i years
PAi36	Total abstract words of valid invention publications for i years
PAi37	Total abstract words of valid utility model grants for i years
PAi38	Total abstract words of valid invention grants for i years
PAi39	Average abstract words of valid invention publications for i years
PAi40	Average abstract words of valid utility model grants for i years
PAi41	Average abstract words of valid invention grants for i years
PAi45	All valid patent counts for i years
PAi46	Proportion of valid invention publications in all invention publications for i years
PAi47	Proportion of valid utility model grants in all utility model grants for i years
PAi48	Proportion of valid design grants in all design grants for i years
PAi49	Proportion of valid patents in all invention grants for i years
PAi50	Average lifespan of valid invention publications for i years
PAi51	Average lifespan of valid utility model grants for i years
PAi52	Average lifespan of valid design grants for i years
PAi53	Average lifespan of valid invention grants for i years
PAi54	Total backward patent citation counts of valid invention grants for i years
PAi55	Proportion of inventions publication patents in all valid patents for i years

Table 2: Continued.

Valid patent indicator ($i = 1 \sim 10$)	Indicator definition
PAi56	Proportion of utility model grants in all valid patents for i years
PAi57	Proportion of design grants in all valid patents for i years
PAi58	Proportion of inventions grants in all valid patents for i years
PAi59	Total forward patent citation counts of valid patents for i years
PAi60	Total backward nonpatent citation counts for valid invention grants for i years

3.6. Patent Prediction Equation. The patent prediction equation for each of the financial indicators is generated via the dynamic time series forecast model as follows:

$$y_t = y_{t-4} + \sum_{i=1}^{n} c_i x_{i,t-4} + e_t, \quad (1)$$

wherein -4 is Lag = 4; y_t represents each of the financial indicators applied as the dependent variable; and $x_{i,t-4}$ are the PKIs applied as the dependent variables for Lag = 4, while in the F-test, the requirement $p^* < 0.05$ if satisfied e_t is the error term.

Though we found the PKI for Lag = 1 to 4 quarters, the patent prediction equation is used to generate the predictive value for each of the financial indicators after four quarters. Earlier when we communicated with financial investment institutions, they advised us that an investment must last for at least one year because short-term investment for speculation is not the reasonable investment behavior. Therefore, Lag = 4 is used in the patent prediction equation in this study.

3.7. Research Steps. The analysis in this study was conducted through the following steps:

(1) Collecting effective samples with their financial indicators and processing the corresponding valid patent indicators in the course of panel data modeling

(2) Using Granger causality test to obtain PLIs and PKIs for each of the financial indicators

(3) Obtaining patent prediction equations for each of the financial indicators using PKIs and the dynamic time series forecast model

(4) Discussing the performance of investment portfolios selected by predictive values and predictive growth rates of various financial indicators and comparing with the market trend

4. Result and Discussion

4.1. Patent Leading Indicator. Table 3 shows the statistics derived from PLI and PKI analysis. For financial indicators of the stock price, ROA, and ROE, not all 570 patent indicators are statistically significant for the prediction. However, we find the PLIs that are statistically significant for predicting each of the financial indicators, namely, the stock price, ROA, and ROE. PKIs also exist for every financial indicator. Details of PKIs are shown in Table 4.

Table 3: Number of PLIs and PKIs for financial indicators.

Financial indicator	PLI & PKI	Lag			
		Lag = 1	Lag = 2	Lag = 3	Lag = 4
Stock price	Number of PLIs	369	408	127	74
	Number of PKIs	29			
ROA	Number of PLIs	305	258	162	148
	Number of PKIs	92			
ROE	Number of PLIs	165	107	95	89
	Number of PKIs	56			

For predicting the stock price, there are a number of PLIs for each of the lags. Among these, the one with the largest number of PLIs is Lag = 2 and it has 408 PLIs. On the other hand, Lag = 4 has 74 PLIs, which is the least number. The number of PKIs is 29. When considering the data collection interval, the longest interval is one year and it has 10 PKIs, and the least is the six-year interval, which has one PKI. As regards patent features, the largest number of PKIs, 10 (PA160~PAX60), is related to "total backward nonpatent citation counts for valid invention grants." The next is "total independent claim counts of valid invention publications" with 4 PKIs (PA324, PA424, PA524, and PA724).

For predicting ROA, each of the lags has PLIs. The largest number, 305 PLIs, is for Lag = 1, and the least number, 148 PLIs, is for Lag = 4. There are 92 PKIs, the data collection interval, of which the most, 11 PKIs each, are for one-year and two-year intervals. The least number of PKIs, 6, is for the six-year interval. As regards the patent features, the largest number of 10 PKIs (PA130~PAX30) relates "total drawing counts of valid invention publications" and "average drawing counts of valid invention publications" (PA133~PAX33). 9 PKIs (PA160~PA760, PA960~PAX60) relate to "total backward nonpatent citation counts for valid invention grants."

For predicting ROE, each of the lags has PLIs. The largest number, 165 PLIs, is for Lag = 1, and the least, 89 PLIs, is for Lag = 4. There are 56 PKIs found. Considering the data collection interval, the most, 8 PKIs, are with the two-year interval, while the least number of PKIs, 5, is with one-year, four-year, five-year, six-year, eight-year, nine-year, and ten-year intervals. As for patent features, the most, 10 PKIs (PA135~PAX35), relate to two features, "average drawing counts of valid invention grants" and (PA160~PAX60) "total backward nonpatent citation counts for valid invention grants." Next, 9 PKIs (PA248~PAX48) relate to "proportion of valid design grants in all design grants" and 9 PKIs (PA252~PAX52) relate also to "average lifespan of valid design grants".

Table 4: PKIs for financial indicators.

Financial indicator	PKI
Stock price	PA104, PA107, PA108, PA114, PA120, PA126, PA132, PA138, PA154, PA160, PA259, PA260, PA324, PA360, PA404, PA412, PA424, PA454, PA460, PA524, PA560, PA660, PA701, PA724, PA760, PA860, PA960, PAX59, PAX60
ROA	PA106, PA107, PA109, PA110, PA130, PA131, PA132, PA133, PA134, PA135, PA160, PA230, PA231, PA232, PA233, PA234, PA235, PA246, PA248, PA252, PA259, PA260, PA330, PA332, PA333, PA334, PA335, PA346, PA348, PA352, PA359, PA360, PA403, PA425, PA430, PA432, PA433, PA435, PA448, PA452, PA460, PA503, PA530, PA533, PA534, PA535, PA548, PA552, PA560, PA630, PA633, PA635, PA648, PA652, PA660, PA711, PA730, PA733, PA734, PA735, PA748, PA752, PA758, PA760, PA811, PA830, PA833, PA834, PA835, PA848, PA852, PA858, PA903, PA911, PA930, PA933, PA934, PA935, PA948, PA952, PA958, PA960, PAX03, PAX11, PAX30, PAX33, PAX34, PAX35, PAX48, PAX52, PAX58, PAX60
ROE	PA107, PA109, PA110, PA135, PA160, PA231, PA233, PA234, PA235, PA248, PA252, PA259, PA260, PA333, PA335, PA346, PA348, PA352, PA359, PA360, PA433, PA435, PA448, PA452, PA460, PA533, PA535, PA548, PA552, PA560, PA633, PA635, PA648, PA652, PA660, PA733, PA735, PA748, PA752, PA758, PA760, PA833, PA835, PA848, PA852, PA860, PA933, PA935, PA948, PA952, PA960, PAX33, PAX35, PAX48, PAX52, PAX60

Regarding the lag, the most PLIs usually exist for Lag = 1, except for the stock price. As the lag increases, the number of PLIs tends to decrease.

Regarding PKIs, they are present for all financial indicators, of which the largest number, 92 PKIs, relates to ROA and the least, 29 PKIs, relates to stock price. The number of PKIs in each data collection interval is shown in Figure 1. The one-year interval has 26 PKIs, the largest number, and the two-year interval has 21 PKIs, the next largest number. As the data collection interval increases, the number of PKIs tends to decrease.

For different financial indicators, PKIs change and are not all the same. Figure 2 shows the top PKI statistics. Among all PKIs, most of them are relating to "total number of nonpatent backward citations of patents for valid invention grants (PAi60)" of 29 PKIs; the second most are relating to "average drawing counts of valid invention grants (PAi35)" of 20 PKIs; the third most are relating to "average drawing counts of valid invention publications" of 19 PKIs.

4.2. Patent Prediction Equation. Though lots of PKIs are found, the positive and/or negative contribution to financial indicators are not yet revealed. We, therefore, construct the patent prediction equations for quantitatively modeling the relationship.

The patent prediction equation is constructed by four predicting modeling periods, namely, period I : 2016Q4 predicting 2017Q4; period II : 2017Q1 predicting 2018Q1; period III : 2017Q2 predicting 2018Q2; period IV : 2017Q3 predicting 2018Q3.

Details of patent prediction equations are shown in Tables 5–7, though some of them are provided with poor goodness of fit.

For predicting the stock price, the patent prediction equation is shown in Table 5, in which PRICE represents the stock price and (−4) represents Lag = 4. The adjusted $R^2 = 0.6568$ shows that though the patent prediction equation is not perfect, it might have some prediction capability. The number of PKIs in Table 3 is 29, here only 7 PKIs left in Table 5 because they are linearly combined in the patent prediction equation, and each of the PKIs must satisfy significance $p^* < 0.05$.

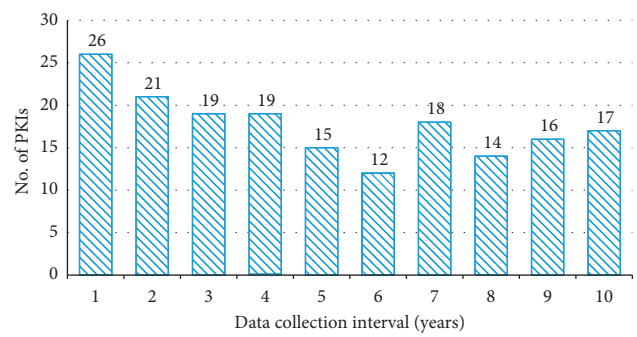


Figure 1: The number of PKIs in each data collection interval.

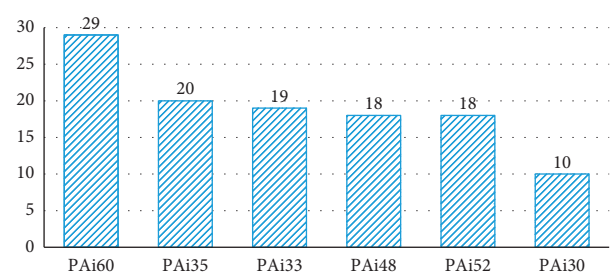


Figure 2: Top PKIs for financial indicators.

For predicting ROA, the patent prediction equation with the adjusted $R^2 = 0.3339$ is shown in Table 6. The patent prediction equation might not have enough prediction capability. The number of PKIs in Table 3 is 92, here only 26 PKIs left in Table 6.

For predicting ROE, the patent prediction equation with the adjusted $R^2 = 0.2606$ is shown in Table 7. The prediction capability might be not good. The number of PKIs in Table 3 is 56, here only 9 PKIs left in Table 7.

From the above analysis, it is possible to construct patent prediction equations for Lag = 4 for predicting the stock price, ROA, and ROE. In terms of the goodness of fit, the adjusted R^2, that of the stock price patent prediction equation at 0.6568 is the best; the worst is of the ROE patent prediction equation at 0.2606. The most PKIs included, numbering 26, are in the ROA patent prediction equation;

Table 5: Patent prediction equation for the stock price.

Dependent var.	PRICE			
Independent var.	Coefficient	Std. error	t-statistic	p
C	−5.8913	0.4060	−14.5118	0.0000***
PRICE(−4)	0.9106	0.0071	128.5839	0.0000***
PA104(−4)	0.9661	0.3327	2.9038	0.0037**
PA108(−4)	−0.5771	0.2213	−2.6075	0.0091**
PA160(−4)	0.5419	0.1785	3.0361	0.0024**
PA424(−4)	0.4124	0.1666	2.4749	0.0133*
PA454(−4)	−0.7103	0.1715	−4.1411	0.0000***
PA960(−4)	−2.8174	1.0804	−2.6078	0.0091**
PAX60(−4)	3.7089	1.0664	3.4780	0.0005***
Patent prediction equation	PRICE = −5.8913 + 0.9107 ∗ PRICE(−4) + 0.9663 ∗ PA104(−4) − 0.5771 ∗ PA108(−4) + 0.5419 ∗ PA160(−4) + 0.4124 ∗ PA424(−4) − 0.7103 ∗ PA454(−4) − 2.8174 ∗ PA960(−4) + 3.7089 ∗ PAX60(−4)			
Adjusted R^2	0.6568	p (F-statistic)		0.0000***

$p^* < 0.05$, $p^{**} < 0.05$, and $p^{***} < 0.05$.

Table 6: Patent prediction equation for ROA.

Dependent var.	ROA			
Independent var.	Coefficient	Std. error	t−statistic	p
C	0.8407	0.0430	19.5351	0.0000***
ROA(−4)	0.5373	0.0087	62.0963	0.0000***
PA106(−4)	−0.0279	0.0090	−3.0935	0.0020**
PA107(−4)	−0.0514	0.0126	−4.0975	0.0000***
PA130(−4)	0.0255	0.0111	2.3012	0.0214*
PA131(−4)	0.0610	0.0152	4.0166	0.0001***
PA134(−4)	−0.0526	0.0203	−2.5951	0.0095**
PA160(−4)	0.0275	0.0093	2.9488	0.0032**
PA330(−4)	−0.0613	0.0223	−2.7423	0.0061**
PA359(−4)	0.0448	0.0104	4.2888	0.0000***
PA425(−4)	−0.0252	0.0122	−2.0702	0.0385∗
PA530(−4)	0.1066	0.0376	2.8354	0.0046**
PA535(−4)	−0.1164	0.0531	−2.1898	0.0286*
PA548(−4)	−0.1872	0.0525	−3.5667	0.0004***
PA660(−4)	0.0183	0.0078	2.3495	0.0188*
PA734(−4)	0.3742	0.1443	2.5933	0.0095**
PA735(−4)	0.3275	0.1290	2.5392	0.0111*
PA811(−4)	0.0575	0.0226	2.5455	0.0109*
PA852(−4)	−0.0186	0.0084	−2.2196	0.0265*
PA858(−4)	−2.4780	0.8502	−2.9146	0.0036**
PA903(−4)	0.0215	0.0069	3.1319	0.0017**
PA930(−4)	−0.1071	0.0325	−3.2966	0.0010**
PA933(−4)	1.3999	0.3236	4.3265	0.0000***
PA934(−4)	−0.3665	0.1448	−2.5303	0.0114*
PAX33(−4)	−1.3631	0.3249	−4.1956	0.0000***
PAX35(−4)	−0.2616	0.1159	−2.2565	0.0241∗
PAX48(−4)	0.1768	0.0574	3.0811	0.0021**
PAX58(−4)	2.6567	0.8520	3.1180	0.0018**
Patent prediction equation	ROA = 0.8407 + 0.5373 ∗ ROA(−4) − 0.0279 ∗ PA106(−4) − 0.0514 ∗ PA107(−4) + 0.0255 ∗ PA130(−4) + 0.0610 ∗ PA131(−4) − 0.0526 ∗ PA134(−4)+0.0275∗PA160(−4) − 0.0613 ∗ PA330(−4) + 0.0448 ∗ PA359(−4) − 0.0252 ∗ PA425(−4) + 0.1066 ∗ PA530(−4) − 0.1164 ∗ PA535(−4) − 0.1872 ∗ PA548(−4) + 0.0183 ∗ PA660(−4) + 0.3742 ∗ PA734(−4) + 0.3275∗PA735(−4) + 0.0575 ∗ PA811(−4) − 0.0186 ∗ PA852(−4) − 2.4780 ∗ PA858(−4) + 0.0215 ∗ PA903(−4) − 0.1071 ∗ PA930(−4) + 1.3999∗PA933(−4) − 0.3665 ∗ PA934(−4) − 1.3631 ∗ PAX33(−4)− 0.2616 ∗ PAX35(−4) + 0.1768 ∗ PAX48(−4) + 2.6567 ∗ PAX58(−4)			
Adjusted R^2	0.3339	p (F−statistic)		0.0000***

$p^* < 0.05$, $p^{**} < 0.01$, and $p^{***} < 0.001$.

Table 7: Patent prediction equation for ROE.

Dependent var.	ROE			
Independent var.	Coefficient	Std. error	t-statistic	p
C	1.0292	0.0358	28.7224	0.0000^{***}
ROE(−4)	0.4873	0.0090	54.0912	0.0000^{***}
PA110(−4)	−0.0736	0.0244	−3.0222	0.0025^{**}
PA160(−4)	0.0540	0.0147	3.6760	0.0002^{***}
PA233(−4)	−0.0311	0.0156	−1.9930	0.0463^{*}
PA260(−4)	−0.0443	0.0185	−2.3888	0.0169^{**}
PA359(−4)	0.0210	0.0089	2.3668	0.0180^{*}
PA552(−4)	−0.0232	0.0060	−3.8484	0.0001^{***}
PA560(−4)	0.0341	0.0127	2.6923	0.0071^{**}
PA735(−4)	0.5444	0.1445	3.7663	0.0002^{***}
PAX35(−4)	−0.6014	0.1455	−4.1332	0.0000^{***}
Patent prediction equation	ROE = 1.0292 + 0.4873 ∗ ROE(−4) − 0.0736 ∗ PA110(−4) + 0.0540 ∗ PA160(−4) − 0.0311 ∗ PA233(−4) − 0.0443 ∗ PA260(−4) + 0.0210 ∗ PA359(−4) − 0.0232 ∗ PA552(−4) + 0.03406 ∗ PA560(−4) + 0.5444 ∗ PA735(−4) − 0.6014 ∗ PAX35(−4)			
Adjusted R^2	0.2606	p (F−statistic)		0.0000^{***}

$p^{*} < 0.05$, $p^{**} < 0.01$, and $p^{***} < 0.001$.

the stock price patent prediction equation has the least, only seven. However, the number of PKIs does not seem to be relevant to the adjusted R^2.

Since the adjusted R^2 of the aforementioned patent prediction equations is between 0.2606 and 0.6568, it might be inappropriate to use any patent prediction equation for predicting any specific financial indicator. However, the predictive values could be applied for constructing specific investment portfolios consisting of some specific stocks.

4.3. Investment Portfolio Performance. The main objective of patent analysis and prediction is to understand how consideration of patent informatics while investing can be commercially beneficial. In this section, because patent prediction equations for financial indicators are established, stock selection criteria based on these patent prediction equations and the performance of investment portfolios are discussed. Since four quarters, that is, Lag = 4, are applied in patent prediction equations, the stock price return rates of all investment portfolios are compared at annual intervals.

Because the following four predictive modeling periods are used in constructing the patent prediction equations, the investment portfolio performance is also observed over these four predicting modeling periods:

Period I : 2016Q4 to 2017Q4.

Period II : 2017Q1 to 2018Q1.

Period III : 2017Q2 to 2018Q2.

Period IV : 2017Q3 to 2018Q3.

Then, we make an objective comparison between the performance of all the A-shares, which represents the market trend and performance of all the effective samples of stocks over the same periods.

4.3.1. Investment Strategy I. Top 100, top 200, and top 300 stocks to be included in the investment portfolios are selected by the higher predictive values of financial indicators. Patent prediction equations are used to generate the predictive values of the selected stocks for each financial indicator in the first quarter of each period. These predictive values are then compared with the averages of the rates on annual return on the actual stock prices.

The above comparisons are shown in Table 8. Because of the decline in the overall economic environment, stock price annual return rates of the A-shares from period I to period IV are negative. Especially, due to the impact of the China-US trade conflict, the decline in the periods III and IV is steeper. However, the performance of all effective samples in each period is better on average by 1.14% than the performance of all A-shares together. It means that the average performance of stocks with patents is higher than the stocks without patents. Thus, patent-based stock selection results in better returns.

Among the investment portfolios selected by the higher predictive stock price, in the four periods, PRICE100 has the best performance in two periods (periods III and IV) and PRICE300 has the best performance in two periods (periods I and II). Of these two, PRICE100 is preferable, going by its better average during the four periods. However, PRICE100 has still 4.59% less average than the A-shares' average and 5.73% less average than the effective samples' average. Therefore, selecting the stocks by the higher predictive stock price seems not to be a good stock selection criterion.

As for the selection of stocks for investment portfolios by the higher predictive ROA, among four periods, ROA100 has the best performance in three periods (periods I, II, and III) and ROA200 has the best performance in one period (period IV). Compared to the all of A-shares and all effective samples, each of the ROA100, ROA200, and ROA300 is better. Among these, ROA100 is preferable for its better average over the four periods and has 15.50% higher average than A-shares' average.

Among the investment portfolios selected by the higher predictive ROE, ROE100 has the best performance in three periods (periods I, II, and III) and ROE300 has the best performance in one period (period IV). Compared with the

Table 8: Performance comparison of investment strategy I.

Investment portfolio	Actual stock price annual return rate				
	Period I 2016Q4⟶2017Q4	Period II 2017Q1⟶2018Q1	Period III 2017Q2⟶2018Q2	Period IV 2017Q3⟶2018Q3	Average
A-shares	−19.42%	−22.77%	−27.85%	−36.86%	−26.73%
All effective samples	−18.12%	−21.51%	−26.94%	−35.79%	−25.59%
PRICE100	−43.16%	−32.62%	−18.29%	−31.19%	−31.32%
PRICE200	−40.08%	−33.18%	−24.06%	−33.86%	−32.80%
PRICE300	−35.92%	−30.16%	−25.06%	−34.72%	−31.47%
ROA100	−1.45%	−8.31%	−10.14%	−24.99%	−11.22%
ROA200	−9.37%	−13.52%	−13.59%	−24.55%	−15.26%
ROA300	−11.45%	−12.74%	−18.02%	−28.16%	−17.59%
ROE100	11.21%	−2.25%	−11.89%	−26.77%	−7.43%
ROE200	2.93%	−4.02%	−14.52%	−27.22%	−10.71%
ROE300	−1.46%	−7.84%	−15.76%	−26.55%	−12.90%

PRICE100, PRICE200, and PRICE300 stand for investment portfolios of top 100, top 200, and top 300 stocks selected by the higher predictive stock price. ROA100, ROA200, and ROA300 stand for top stocks selected by the higher predictive ROA. ROE100, ROE200, and ROE300 stand for top stocks selected by the higher predictive ROE.

A-shares, each effective sample of the ROE100, ROE200, and ROE300 is better. Among the effective samples, ROE100 is preferable for the average of four periods, which has 19.30% higher average than the A-shares' average.

Figure 3 shows the stock performance of all effective samples based on higher predictive values and the investment portfolios preferable over the A-shares' average, wherein E. samples stands for all effective samples. For clear comparison, the performance of A-shares is set at zero on the horizontal axis. The positive values show performance better than the market trend (performance of A-shares), and the negative values show performance worse than the market trend. In Figure 3, the portfolios selected by the higher predictive stock price have bad performance, but the portfolios selected by the higher predictive ROA and the higher predictive ROE have better performance, and ROE100 is the best. ROE100 has 19.30% higher average than the A-shares' average and 18.16% higher average than all effective samples' average.

Let us focus on the periods III and IV in Table 8, though the stock price annual return rates are deeply in the negative due to the impact of the China-US trade conflict. By either of the criteria, ROA100 to ROA300 and ROE100 to ROE300 demonstrate higher performance than the A-shares and among the effective samples. Using the higher predictive ROA and the higher predictive ROE is a good investment strategy for choosing good investment portfolios during the China-US trade conflict.

Meanwhile, for the number of stocks in the investment portfolio, the top 100 is usually the best; as the number of stocks increases, the performance of the investment portfolio tends to decrease. Therefore, it seems advisable to limit the number of stocks in the investment portfolio to less than 200.

4.3.2. Investment Strategy II. For investment strategy II, the stocks in investment portfolios are selected by the higher predictive growth rates of the financial indicators. The patent prediction equations are applied to generate predictive values of the stock price, ROA, and ROE in the first quarter of each period. When compared with real values of the stock price, ROA, and ROE in the first quarter of each period, the corresponding predictive growth rates of the stock price, ROA, and ROE are arrived at.

Top 100, top 200, and top 300 stocks selected by the higher predictive growth rates are set as the investment portfolios, and then the averages of actual stock price annual return rates are examined. The comparison is shown in Table 9.

Table 9 shows that, of the investment portfolios selected by the higher predictive stock price growth rate in the four periods, PRICE100R has the best performance in two periods (periods I and IV) and PRICE200R has the best performance in two periods (periods II and III). Compared with the A-shares, each of the PRICE100R, PRICE200R, and PRICE300R is better. Among these, PRICE200R is preferable for the average of four quarters and has 18.17% better average than the A-shares' average.

Among the investment portfolios selected by the higher predictive ROA growth rate, ROA300R has the best performance in three periods (periods II, III, and IV) and ROA100R has the best performance in one period (period I). Compared with the A-shares, all three, ROA100R, ROA200R, and ROA300R, are better, and among the three, ROA300R is preferable for the better average of four quarters and has 2.66% better average than the A-shares' average.

Among the investment portfolios selected by the higher predictive ROE growth rate, ROE200R has the best performance in two periods (periods I and III), ROE100R performed best in period IV, and ROE.300 has the best performance in one period (period II). Compared with the A-shares, each of the three effective samples is better than the A-shares' average. Among these, ROE200R is preferable for the best average of four quarters and has 1.98% better average than the A-shares' average.

With the higher predictive financial indicator growth rate as a stock selection criterion, Figure 4 shows the stock performance of all effective samples and the preferable investment portfolios selected by the higher predictive financial indicator growth rate when compared with the

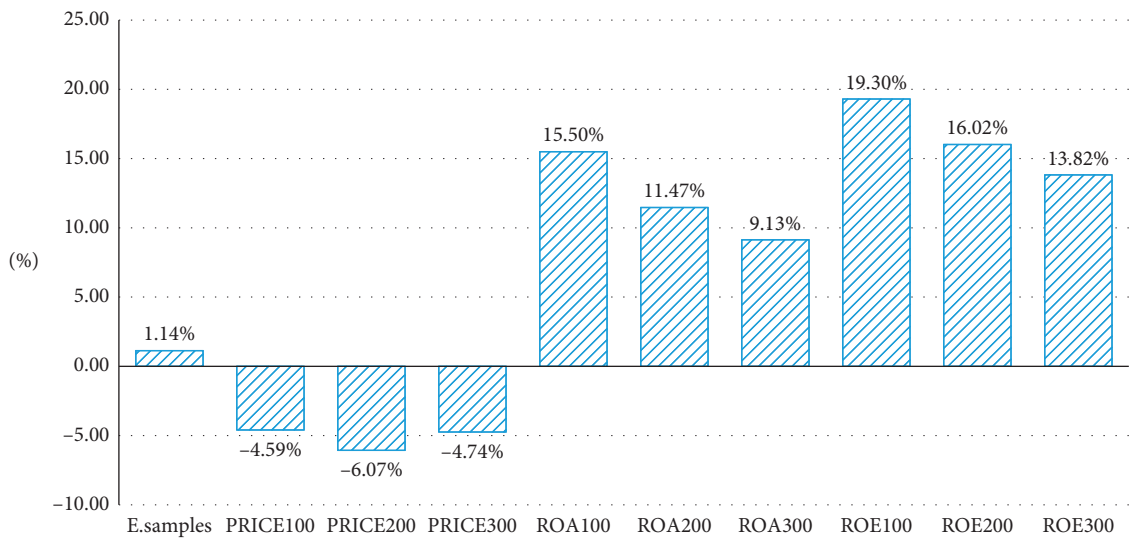


Figure 3: Performance comparison of investment strategy I.

Table 9: Performance comparison of investment strategy II.

Investment portfolio	Actual stock price return rate				Average
	Period I 2016Q4⟶2017Q4	Period II 2017Q1⟶2018Q1	Period III 2017Q2⟶2018Q2	Period IV 2017Q3⟶2018Q3	
A-shares	−19.42%	−22.77%	−27.85%	−36.86%	−26.73%
All effective samples	−18.12%	−21.51%	−26.94%	−35.79%	−25.59%
PRICE100R	5.02%	−7.23%	−15.93%	−19.97%	−9.53%
PRICE200R	4.16%	−1.98%	−14.30%	−22.09%	−8.55%
PRICE300R	−0.26%	−6.08%	−15.04%	−24.35%	−11.43%
ROA100R	−10.90%	−21.28%	−30.89%	−37.14%	−25.05%
ROA200R	−13.98%	−22.35%	−27.84%	−36.06%	−25.06%
ROA300R	−13.64%	−20.03%	−27.26%	−35.35%	−24.07%
ROE100R	−16.01%	−23.31%	−27.94%	−36.29%	−25.89%
ROE200R	−12.79%	−21.16%	−27.46%	−37.58%	−24.75%
ROE300R	−15.83%	−21.00%	−28.04%	−36.88%	−25.44%

PRICE100R, PRICE200R, and PRICE300R stand for investment portfolios of top 100, top 200, and top 300 stocks selected by the higher predictive stock price growth rate. ROA100R, ROA200R, and ROA300R stand for top stocks selected by the higher predictive ROA growth rate. ROE100R, ROE200R, and ROE300R stand for top stocks selected by the higher predictive ROE growth rate.

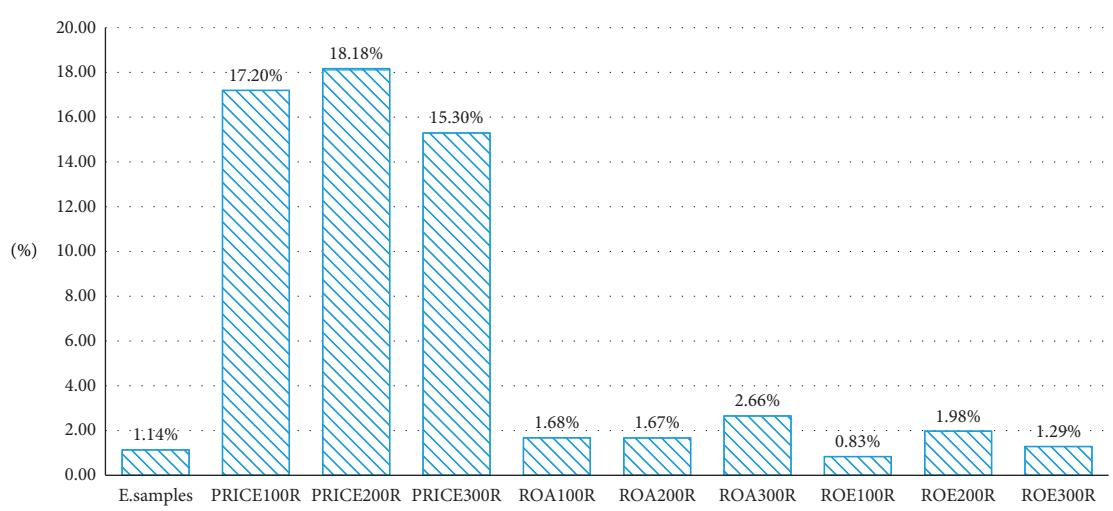


Figure 4: Performance comparison of investment strategy II.

A-shares' average. For a clear comparison, the performance of A-shares is placed at zero on the horizontal axis. The positive value means better performance than the market trend, and the negative value means performance worse than the market trend.

In Figure 4, all investment portfolios are seen to have better performance than the market trend. Especially, PRICE100R, PRICE200R, and PRICE300R are outstanding, and PRICE200R is the best. PRICE200R is 18.18% higher than the A-shares' average and 17.04% higher than all effective samples' average.

Let us focus on periods III and IV in Table 9 above, though the stock price annual return rates are seriously negative due to the impact of the China-US trade conflict. Any of PRICE100R to PRICE300R has higher performance than the A-shares and all effective samples. The higher predictive stock price growth rate is also a good investment strategy and useful in identifying good stocks for inclusion in investment portfolios during the China-US trade conflict. However, when compared with Figure 3, we find PRICE200R is 1.12% less than ROE100; therefore, ROE100 is the best investment portfolio in this study.

In Figure 4, going by the number of stocks in the investment portfolio, the top 200 stocks have shown better performance by two of the three criteria; though as the number of stocks increases, the performance of the investment portfolio tends to decrease.

5. Conclusion and Recommendation

Based on patent data, the stock price, ROA, and ROE of China A-shares from 2016Q4 to 2018Q3, we constructed algorithms for finding PLIs, PKIs, patent prediction equations, and strategies for choosing stocks for investment. We arrived at the following conclusions:

(1) While Chen et al. proposed the use of PLIs for the stock price prediction [3], this study found that PLIs are useful in predicting ROA and ROE.

(2) The number of PLIs was the largest for Lag = 1, that is, one-quarter lead. As the lag increased, the number of PLIs tended to decrease.

(3) This study also found that PKIs are useful in predicting for all lead periods (Lag = 1 to 4). The number of PKIs for ROA was the largest at 92 and the least at 29 for the stock price.

(4) PKIs for different financial indicators were not all the same, but among them, "total backward nonpatent citation counts for valid invention grants (PAi60)" showed up most frequently. It indicated that patents resulting from the fundamental research might have a strong relationship with a company's financial performance.

(5) This study also constructed patent prediction equations for predicting the stock price, ROA, and ROE. Among them, the ROA patent prediction equation had the most PKIs and the stock price patent prediction equation had preferable goodness of fit with the adjusted $R^2 = 0.6568$. The number of PKIs in the patent prediction equation did not show relevance to the goodness of fit.

(6) The stock portfolios selected by the higher predictive ROA, the higher predictive ROE, and the higher predictive stock price growth rate had higher performance than the market trend, wherein ROE100 was the best. Even under the impact of the China-US trade conflict, any of ROA100 to ROA300, ROE100 to ROE300, and PRICE100R to PRICE300R showed better performance than the A-shares' average.

(7) Chen et al. [3] proposed an investment strategy to select stocks by the higher predictive stock price growth rate. This study found alternative investment strategies to select stocks by the higher predictive ROA and the higher predictive ROE, wherein ROE100 had the best performance among all the stock portfolios.

(8) The patent-based prediction algorithm proposed in this study was proved useful for identifying good stocks of inclusion in investment portfolios, despite the overall economic environment being unstable and the worsened decline due to the China-US trade conflict. It is believed that the proposed algorithm is also useful for choosing stocks during the COVID-19 pandemic. However, to test that speculation, more data are required.

This study raises some possible issues for further research. For example, how to appropriately combine three preferable stock strategies, the predictive ROA, the predictive ROE, and the predictive stock price growth rate, to get the best performance? And, can AI be used for modeling patent prediction equations and compiling investment portfolios that would perform better?

Data Availability

The data, which were provided by Shenzhen TekGlory Intellectual Property Data Technologies, Ltd., supporting the conclusions of this study are not available for the readers because of confidentiality and commercial reasons.

Conflicts of Interest

None of the authors have any conflicts of interest.

Acknowledgments

The authors would like to thank Shenzhen TekGlory Intellectual Property Data Technologies, Ltd. for processing all the valid patent indicators of China A-shares.

References

[1] Z. Deng, B. Lev, and F. Narin, "Science and technology as predictors of stock performance," *Financial Analysts Journal*, vol. 55, no. 3, pp. 20–32, 1999.

[2] P. Thomas, "A relationship between technology indicators and stock market performance," *Scientometrics*, vol. 51, no. 1, pp. 319–333, 2001.
[3] T.-M. Chen, C.-C. Wei, and H.-C. Che, "Contribution of patent indicators to China stock performance," in *Proceedings of the 7th International Congress on Advanced Applied Informatics (ICAAI 2018)*, pp. 793–798, Yonago, Japan, July 2018.
[4] B. Branch, "Research and development activity and profitability: a distributed lag analysis," *Journal of Political Economy*, vol. 82, no. 5, pp. 999–1011, 1974.
[5] Z. Griliches, "Market value, R&D, and patents," *Economics Letters*, vol. 7, no. 2, pp. 183–187, 1981.
[6] I. Cockburn and Z. Griliches, "Industry effects and appropriability measures in the stock M," *The American Economic Review*, vol. 78, no. 2, p. 419, 1988.
[7] B. H. Hall, A. Jaffe, and M. Trajtenberg, "Market value and patent citations," *RAND Journal of Economics*, vol. 36, no. 1, pp. 16–38, 2005.
[8] B. Branch and C. Chichirau, "Mispricing vs risk premia in R&D-intensive firms," *International Review of Financial Analysis*, vol. 19, no. 5, pp. 358–367, 2010.
[9] M. M. Crossan and M. Apaydin, "A multi-dimensional framework of organizational innovation: a systematic review of the literature," *Journal of Management Studies*, vol. 47, no. 6, pp. 1154–1191, 2010.
[10] S. Pandit, C. E. Wasley, and T. Zach, "The effect of research and development (R&D) inputs and outputs on the relation between the uncertainty of future operating performance and R&D expenditures," *Journal of Accounting, Auditing & Finance*, vol. 26, no. 1, pp. 121–144, 2011.
[11] S. Fabrizi, S. Lippert, P. Norback, and L. Persson, "Venture capital, patenting, and usefulness of innovations," 2011.
[12] D. Hirshleifer, P.-H. Hsu, and D. Li, "Innovative efficiency and stock returns," *Journal of Financial Economics*, vol. 107, no. 3, pp. 632–654, 2013.
[13] T. Caner, O. Bruyaka, and J. E. Prescott, "Flow signals: evidence from patent and alliance portfolios in the US biopharmaceutical industry," *Journal of Management Studies*, vol. 55, no. 2, pp. 232–264, 2018.
[14] G. J. Yu and K. Hong, "Patents and R&D expenditure in explaining stock price movements," *Finance Research Letters*, vol. 19, pp. 197–203, 2016.
[15] H. B. Mama, "Innovative efficiency and stock returns: should we care about nonlinearity?" *Finance Research Letters*, vol. 24, pp. 81–89, 2018.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1730321, 11 pages
https://doi.org/10.1155/2020/1730321

Research Article

An Image Denoising Method Based on BM4D and GAN in 3D Shearlet Domain

Shengnan Zhang,[1] **Lei Wang**,[1] **Chunhong Chang**,[1] **Cong Liu**,[1] **Longbo Zhang**,[1] **and Huanqing Cui**[2]

[1]*School of Computer Science and Technology, Shandong University of Technology, Zibo 255000, China*
[2]*Shandong Key Laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao 266590, China*

Correspondence should be addressed to Lei Wang; wanglei0511@sdut.edu.cn and Huanqing Cui; cuihq@sdust.edu.cn

Received 14 January 2020; Revised 15 March 2020; Accepted 31 March 2020; Published 28 April 2020

Guest Editor: Chunjia Han

To overcome the disadvantages of the traditional block-matching-based image denoising method, an image denoising method based on block matching with 4D filtering (BM4D) in the 3D shearlet transform domain and a generative adversarial network is proposed. Firstly, the contaminated images are decomposed to get the shearlet coefficients; then, an improved 3D block-matching algorithm is proposed in the hard threshold and wiener filtering stage to get the latent clean images; the final clean images can be obtained by training the latent clean images via a generative adversarial network (GAN).Taking the peak signal-to-noise ratio (PSNR), structural similarity (SSIM for short) of image, and edge-preserving index (EPI for short) as the evaluation criteria, experimental results demonstrate that the proposed method can not only effectively remove image noise in high noisy environment, but also effectively improve the visual effect of the images.

1. Introduction

The typical image denoising methods can be commonly classified into three schemes: the filtering-based methods, the decomposition-based methods, and dictionary learning-based methods [1]. The classical filtering includes the median filter [2] and the wiener filter [3]. The basic principle of decomposition-based methods is to decompose the contaminated images into low-pass and high-pass subbands and then separate the image noise by manipulating the obtained coefficients. And, the wavelet transform is the typical decomposition tool. The basic principle of the dictionary learning-based methods is to sparsely represent the noisy image by overcomplete atoms, and only the large representation coefficients are used to reconstruct the original image while the small coefficients are discarded [4–6]. Another famous strategy for denoising is based on the self-similarity of the image, such as the block-matching and 3D filtering (BM3D) algorithm [7]. The above methods usually perform well, but they will lose the edge, texture, and other details and result in blur and block effect when the noise is heavy.

Nowadays, since the adaptive space estimation strategy for nonlocal mean [8]can effectively alleviate the high complexity and low efficiency of nonlocal mean filtering, the nonlocal similarity becomes an effective feature for image denoising [9]. Its main disadvantage, however, is the loss of the directions. So, the geometric regularity of images cannot be effectively captured to sparsely represent the features of the original images [10]. Besides, the computation of the nonlocal similarity is implemented only in the spatial domain. As reported, the multiscale geometric transformations, such as contourlet transform, can greatly help to suppress the heavy noise in the frequency domain [11]. But this scheme is limited to the mathematical properties of the selected transformation. On the other hand, the deep learning technologies, such as the convolutional neural network [12] and the generative adversarial network [13], have achieved great success in the areas of image classification, target recognition [14], and image fusion [15].

Therefore, a novel image denoising method based on block matching with 4D filtering in the 3D shearlet transform domain and the generative adversarial network is proposed. The 3D shearlet transform provides better mathematical properties than the commonly used wavelet or contourlet to capture the anisotropic features of images in different scales and directions. In addition, the traditional BM3D is extended into four 4D space, which can effectively improve the edge and texture details of the images. The output of the BM4D is used as the input to the designed generative adversarial network to make full use of its good learning ability.

The remainder of this paper is organized as follows. The details of the proposed method are presented in Section 2. Experimental results and discussions are shown in Section 3. Finally, the whole paper is concluded in Section 4.

2. Methodology

2.1. The Architecture of Proposed Method. The overall structure of the proposed method is shown in Figure 1. Through multiscale decomposition and directional partition, the 3D shearlet coefficients are obtained. Then, the coefficients are input into the hard threshold and wiener filtering contained in the BM4D model. For the hard threshold, the similar cubes are extracted from volume observation and then they are combined together. If the distance between the cubes is smaller than the setting threshold, the collaborative filters are carried out to apply on the similar cubes. In the aggregation process, the basic volume estimation is generated by the adaptive convex combination. For the process of the wiener filtering, the aggregation is implemented by the inversion of the shearlet transformation. Finally, the clean results are obtained by the designed generative adversarial network.

2.2. The 3D Shearlet Transform. In the 3D space, the shearlet region $P_i\,(i = 1, 2, 3)$ is obtained by combining the function system associated with the pyramid region in the 3D Fourier space R^3.

For $d = 1, 2, 3, l = (l_1, l_2) \in Z^2$, the 3D shear system associated with P_d in the shearlet region is defined as a set:

$$\left\{\psi^d_{j,l,k} = \left|\det A^{(d)}\right|^{j/2}\psi^{(d)}\left(B^{(d)}_l\left(A^{(d)}\right)^j x - k\right):\ j \in Z,\ l = (l_1, l_2) \in \Lambda_j,\ k \in Z^3,\ d = 1, 2, 3,\right\}, \tag{1}$$

where $\psi^{(d)} \in L^2(R^3)$, $\{(l_1, l_2):\ -2^j \le l_1, l_2 \le 2^j\}$, and $\Lambda_j \subset Z^2$. $\mathbf{A}^d$ is the specific anisotropic expansion matrix and $\mathbf{B}^d_l$ is the specific shear matrix. More details can be found in the literature [16, 17]. In Figure 2, a shearlet in 3D space is shown.

2.3. The Improved 3D Block-Matching Algorithm. Let $z: X \longrightarrow R$ be the noise form:

$$z(x) = y(x) + \eta(x), \quad x \in X, \tag{2}$$

where y is the original unknown volume signal, $x \subset Z^3$ is the 3D signal, and $\eta(\cdot) \sim N(0, \sigma^2)$ is an independent and uniformly distributed Gaussian noise, whose standard deviation is noted as σ.

The goal of the improved 3D block-matching algorithm [18]was to obtain the estimation $\hat{y}$ of y from the noise observations. The implementation of the improved algorithm was divided into two cascade stages: the hard threshold and wiener filtering [19], each of which includes 3 steps: grouping, collaborative filtering, and aggregation.

For the hard threshold, let C^z_{xR} represent a cube whose volume is $L \times L \times L$, where $L \in \mathrm{N}$ was extracted from z. The similarity between two cubes is measured by photometric distance:

$$d\left(C^z_{x_i}, C^z_{x_j}\right) = \frac{\left\|C^z_{x_i} - C^z_{x_j}\right\|^2_2}{L^3}, \tag{3}$$

where $\|\cdot\|$ represents the summation of the squared differences between the corresponding intensities of the two cubes and the denominator L^3 is the normalization factor. No prefiltering is performed before cube matching, so the similarity of noise observations can be directly tested.

In the grouping step, cubes similar to each other are extracted from z and combined to form a group for each cube $C^z_{x_R}$. If the distance between two cubes was not larger than the predefined threshold τ^{ht}_{match}, the two cubes are considered to be similar. Similar to $C^z_{x_R}$, we here firstly define a set that contains the indexes for the cubes as follows:

$$S^z_{x_R} = \left\{x_i \in X:\ d\left(C^z_{x_R}, C^z_{x_i}\right) \le \tau^{ht}_{\text{match}}\right\}. \tag{4}$$

Then, a four-dimensional group is built by the above formula:

$$G^z_{S^z_{x_R}} = \coprod_{x_i \in S^z_{x_R}} C^z_{x_i}, \tag{5}$$

where the reference cube (represented by R) matches a set of similar cubes located in the 3D data. Particularly, the coordinates x_R and x_i correspond to the tail and head of the arrow connecting the cubes in formula (4), respectively. Note that since the distance from any cube to itself is always 0, according to the definition of formula (4), each formula (5) must contain at least its reference cube.

In the collaborative filtering step, a joint four-dimensional transformation T^{ht}_{4D} was applied to each dimension of equation (5), respectively. Then, by a hard threshold operator γ^{ht} with the threshold $\sigma\lambda_{4D}$, the obtained four-dimensional group spectrum is

$$\gamma^{ht}\left(T^{ht}_{4D}\left(G^z_{S^z_{x_R}}\right)\right). \tag{6}$$

Representing the filter group, it is transformed into the following form:

$$T^{ht^{-1}}_{4D}\left(\gamma^{ht}\left(T^{ht}_{4D}\left(G^z_{S^z_{x_R}}\right)\right)\right) = \hat{G}^y_{S^z_{x_R}} = \coprod_{x_i \in S^z_{x_R}} \hat{C}^y_{x_i}. \tag{7}$$

For each unknown volume data y, the estimated $\hat{C}^y_{x_i}$ of the original $C^y_{x_i}$ was extracted separately. Formula (7) was an overcompleted representation of the denoising data because

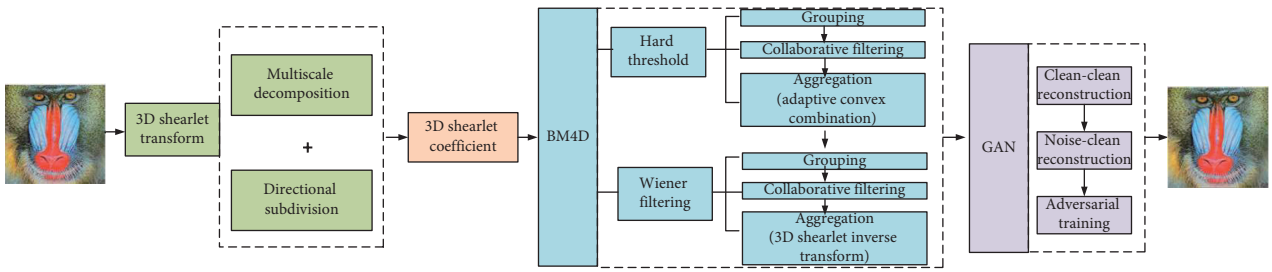


Figure 1: The architecture of the proposed image denoising method.

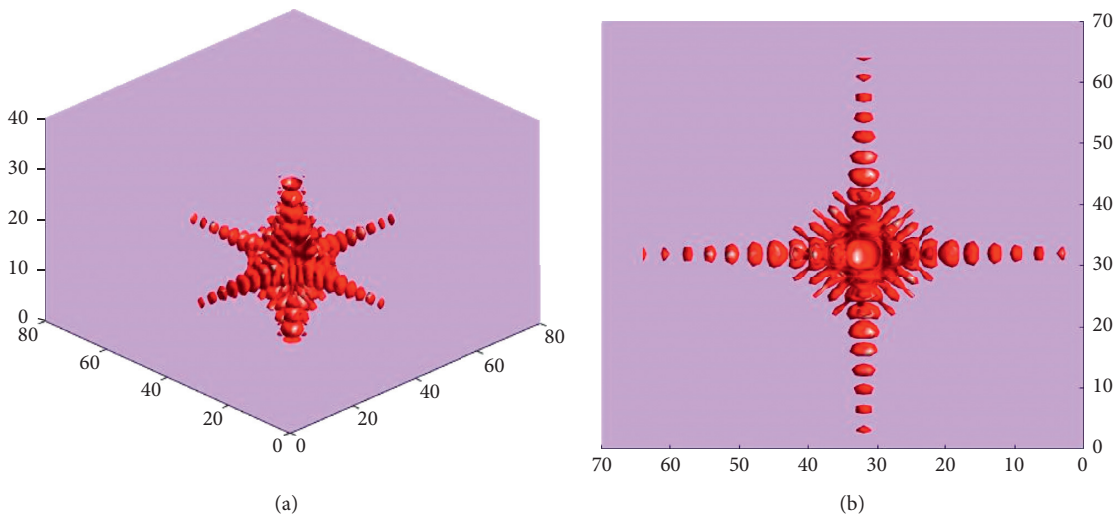


Figure 2: The shearlet in 3D space and its projection. (a) The 3D shearlet and (b) the top view of the 3D shearlet.

there may be overlap between the cubes in the same group and different groups.

In the aggregation step, the redundancy is used to generate a basic volume estimation by adaptive convex combination:

$$\widehat{y} = \frac{\sum_{x_R \in X}\left(\sum_{x_i \in S^z_{x_R}} w^{ht}_{x_R} \widehat{C}^y_{x_i}\right)}{\sum_{x_R \in X}\left(\sum_{x_i \in S^z_{x_R}} w^{ht}_{x_R} \chi_{x_i}\right)}, \tag{8}$$

where $w^{ht}_{x_R}$ is the group-related weight and $\chi_{x_i}\colon X \longrightarrow \{0,1\}$ is the feature (indicator) function of the $\widehat{C}^y_{x_i}$ domain, that is, $\chi_{x_i} = 1$, at the coordinates of $\chi_{x_i} = 0$. The weight is defined as

$$w^{ht}_{x_R} = \frac{1}{\sigma^2 N^{ht}_{x_R}}, \tag{9}$$

where σ is the standard deviation of noise in z and $N^{ht}_{x_R}$ is the number of nonzero coefficients in formula (6). Since the coefficient always remains the same after doing the threshold operation, that is, $N^{ht}_{x_R} \geq 1$, the denominator of equation (9) is never equal to zero. The numerical $N^{ht}_{x_R}$ has two functions: on the one hand, it measures the sparsity of the threshold spectrum in (5) and on the other hand, it approximates the total residual noise variance of the group estimation in (6). As a result, the groups that are in a high degree of correlation will be given more weight, while other groups with larger residual noise are punished with less weight.

For the Wiener filtering, the cube matching is performed within the basic estimation of $\widehat{y}^{ht}$. In fact, since the noise level in $\widehat{y}^{ht}$ is much smaller than that in the noise observation z, it is expected to obtain the more reliable match to make the packet data more sparse. Formally, for each reference cube $\widehat{C}^{y^{ht}}_{x_R}$ extracted from the basic estimated $\widehat{y}^{ht}$, its cube-like coordinate set is constructed as follows:

$$\widehat{S}^{y^{ht}}_{x_R} = \left\{x_i \in X\colon d\left(\widehat{C}^{y^{ht}}_{x_R}, \widehat{C}^{y^{ht}}_{x_i}\right) < \tau^{\text{wie}}_{\text{match}}\right\}. \tag{10}$$

The collaborative filtering here is implemented as an empirical Wiener filter. Similar to formula (6), it firstly uses the coordinate set (9) to extract a set of $\widehat{G}^{y^{ht}}_{\widehat{S}^{y^{ht}}_{x_R}}$ from $\widehat{y}^{ht}$ and then defines the empirical Wiener filter coefficients as

$$W_{\widehat{S}^{y^{ht}}_{x_R}} = \frac{\left|T^{\text{wie}}_{4D}\left(\widehat{G}^{y^{ht}}_{\widehat{S}^{y^{ht}}_{x_R}}\right)\right|^2}{\left|T^{\text{wie}}_{4D}\left(\widehat{G}^{y^{ht}}_{\widehat{S}^{y^{ht}}_{x_R}}\right)\right|^2 + \sigma^2}, \tag{11}$$

where σ is the standard deviation of noise and T^{wie}_{4D} is a transformation operator that is composed of four one-dimensional linear transformations. Such transformations are usually different from those in T^{ht}_{4D}. Subsequently, the same

set of formula (10) is used to extract the second noise group from the observed z, noted as $G^{z}_{\underset{S^{y}_{x_R}}{\frown ht}}$. An element multiplication is implemented between the spectrum of the noise group and the wiener filter coefficient formula (11) as the coefficient shrinkage rate. The group's estimations are

$$\widehat{G}^{y}_{\underset{S^{y}_{x_R}}{\frown ht}} = T_{4D}^{\mathrm{wie}^{-1}}\left(W_{\underset{S^{y}_{x_R}}{\frown ht}} \cdot T_{4D}^{\mathrm{wi^e}}\left(G^{z}_{\underset{S^{y}_{x_R}}{\frown ht}}\right)\right). \tag{12}$$

Then, the inversion of the 3D shearlet transform [14] is applied to shrink the spectrum. The final estimation of $\widehat{y}^{wie}$ is generated by convex combination, which is similar to formula (8), and formula (4) is replaced by formula (10). The aggregation weight of the specific group estimation (11) is defined by the energy of the wiener filter coefficient (12):

$$w^{wie}_{x_R} = \sigma^{-2}\left\|W_{\underset{S^{y}_{x_R}}{\frown ht}}\right\|_2^2. \tag{13}$$

In this way, each formula (13) provides an estimation of the total residual noise variance of the corresponding formula (12).

2.4. The Generative Adversarial Network for Training. After obtaining the intermediate results, we can obtain the final denoising image by training the generative adversarial network (GAN) [20]. The training process is shown in Figure 3.

2.4.1. The Generator Network G. The generated network generates a fake image from the noise image, as shown in Figure 4. The generation network consists of 11 cascaded convolution layers, which are trained to learn the label image and the residual image of the input image. Internal connection is introduced into each block to save information and reduce the training time. In order to maintain good performance and reduce computational complexity, the network adopts a bottleneck structure, in which the number of the first feature mapping, the middle layer, and the last layers are 64 layers. According to suggestion from reference [21, 22], for low-level computer vision problems, a 3×3 convolution kernel is used in each convolution layer and the linear unit (ReLU) is used as the activation function.

2.4.2. The Discriminator Network D. As shown in Figure 5, the discriminator network D is trained to distinguish the fake image and the real image. It has four convolution blocks and two fully connected layers. Each convolution block consists of the convolution layer, the batch normalization layer, and the ReLU activation function. The size of the core K is 3×3, and the number of filters N increases from 64 to 256. The step size S of each convolution layer is 2 to reduce the resolution of the image. The probability that the inputting image is noiseless is generated by a fully connected layer of 1024 neurons.

2.4.3. Adversarial Training. The aim is to use the adversarial strategy to train a model to remove the image noising. Adversarial training is a way to train the generator network $G$to generate samples from real data $x \sim p_{\mathrm{data}}$. The generator is input into a noise variable $z$with a distribution p_Z and then trained to learn the mapping to the data space. The distribution of the generator model is

$$p_g \sim G\left(z;\theta_g\right), \tag{14}$$

where θ_g is the parameter of the generator network. When training a generator, the essentially exception is to maximize the probability that the samples match the data, which can be noted as $p_{\mathrm{data}}(G(z;\theta_g))$.

To guarantee the above probability, the discriminator network D, whose input is the data sample x and the output is the $D(x,\theta_d)$, should learn to distinguish the generated samples from real samples. It must maximize the probability value assigned to the actual data samples and minimize the probability value assigned to the generated samples, that is,

$$\max_{\theta_d} \mathrm{E}_{x\sim p_{\mathrm{data}}}[\log(D(x))] + \mathrm{E}_{z\sim p_Z}[\log(1-D(G(z)))]. \tag{15}$$

Both the generator and discriminator networks are alternately trained, and they try to cheat each other. Finally, when the generator has successfully learned how to generate the samples from p_{data}, the whole process is converged.

In Figure 6, the experiments on four groups of color images show the necessity and effectiveness of using the GAN.

3. Experimental Results and Discussions

In this section, two groups of experiments are implemented to show the performance of the proposed method. The platform is the Dell workstation M4800 with the Intel CPU 2.5 GHz and 32G RAM, operating under Matlab and Python. PSNR [23], SSIM [24], and the edge-preserving index (EPI for short) [25] are used as objective evaluation measurements. PSNR can be computed by the following formula:

$$\mathrm{PSNR}(y,\widehat{y}) = 10\log_{10}\left(\frac{D^2|\widetilde{X}|}{\sum_{x\in\widetilde{X}}(\widehat{y}(x)-y(x))^2}\right), \tag{16}$$

where D is the peak of y, $\widetilde{X} = \{x \in X\colon y(x) > 10\cdot D/255\}$, and $|\widetilde{X}|$ is the base of $\widetilde{X}$.

The SSIM is defined as

$$\mathrm{SSIM}(x,y) = l(x,y)^{\alpha}\cdot c(x,y)^{\beta}\cdot s(x,y)^{\gamma}, \tag{17}$$

where $l(x,y)$, $c(x,y)$, and $s(x,y)$ can be computed as

$$\begin{aligned} l(x,y) &= \frac{2u_x u_y + c_1}{u_x^2 + u_y^2 + c_1},\\ c(x,y) &= \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2},\\ s(x,y) &= \frac{2\sigma_{xy} + c_3}{\sigma_{\mathrm{xy}} + c_3}, \end{aligned} \tag{18}$$

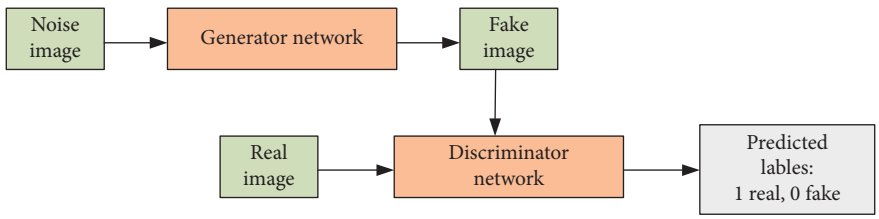


Figure 3: Image denoising by training the GAN.

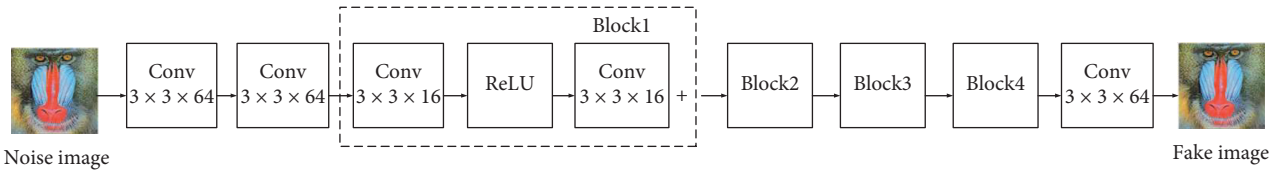


Figure 4: The structure of the generator network.

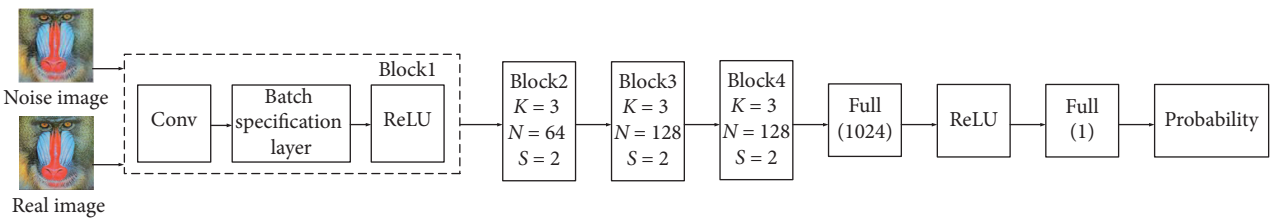


Figure 5: The structure of the discriminator network.

Figure 6: The compared experiments with and without GAN under the Gaussian noise of $\sigma = 25$.

in which x and y are the reference image and the image to be tested, respectively, u_x and $u_y$are the mean values of the two images, σ_x and σ_y are the standard deviation, σ_{xy} is the covariance of x and y, and c_1, c_2, and c_3 are the small constants whose values are positive. It is mainly to avoid the instability when the denominator is 0 in the above formula. When $\alpha = \beta = \gamma = 1$ and $c_3 = c_2/2$, then we can get

$$\mathrm{SSIM}(x, y) = \frac{(2u_x u_y + c_1)(2\sigma_x \sigma_y + c_2)}{(u_x^2 + u_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}. \quad (19)$$

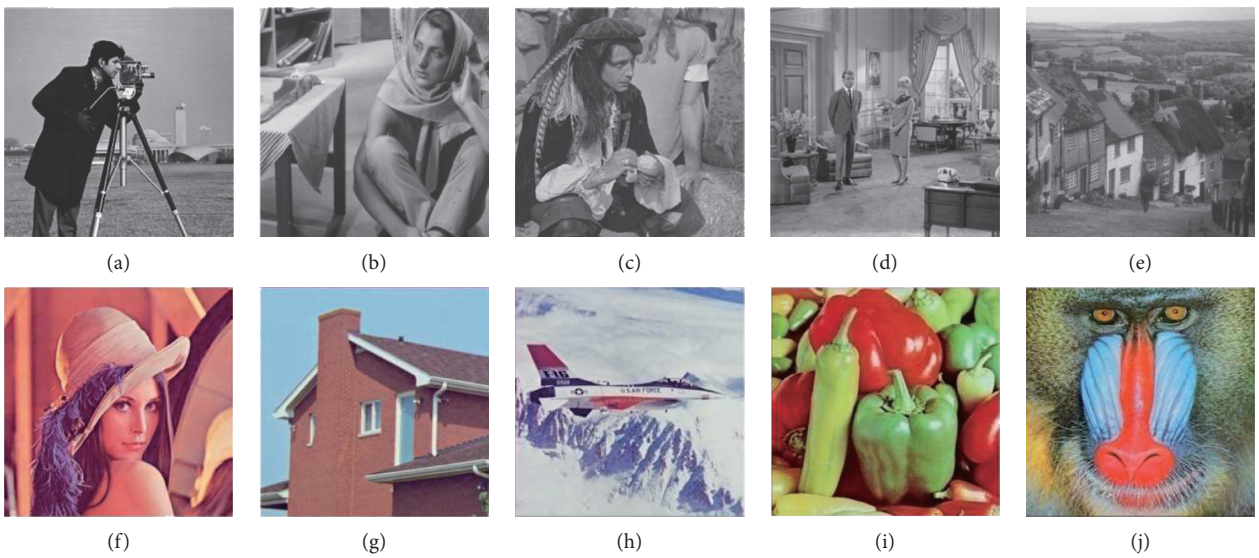


Figure 7: The images used in the first experiment. (a) Cameraman, (b) Barbara, (c) man, (d) couple, (e) hill, (f) Lena, (g) house, (h) F16, (i) peppers, and (j) baboon.

Table 1: The results of the proposed method for Gaussian white noise with different standard deviations.

Image/PSNR	$\sigma = 5$	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$	$\sigma = 25$	$\sigma = 30$	$\sigma = 35$	$\sigma = 40$	$\sigma = 50$
Cameraman	39.10	34.79	33.01	31.43	30.12	29.37	28.69	27.80	26.78
Barbara	38.47	35.08	34.15	32.59	31.27	30.11	29.20	28.05	27.98
Man	38.01	35.49	32.76	31.59	30.30	29.12	28.93	27.81	26.82
Couple	38.58	35.39	33.62	31.36	30.45	29.73	28.99	28.40	27.01
Hill	38.47	34.88	32.90	31.63	30.81	30.16	29.59	28.77	27.96
Lena	38.81	35.99	33.94	33.23	32.27	31.59	30.91	30.11	29.88
House	39.02	36.47	35.17	33.97	33.20	32.68	32.09	31.12	30.79
F16	39.70	36.85	35.39	34.51	33.17	32.01	31.76	30.99	30.51
Peppers	37.39	34.28	33.01	31.91	31.67	30.73	30.88	29.70	29.15
Baboon	35.96	31.29	28.18	27.60	26.16	25.92	25.47	25.01	24.89

Table 2: The results of the different method for standard Gaussian white noise with $\sigma = 15$.

PSNR SSIM EPI Time/s	BM3D	EPLL	TNRD	WNNM	WSNM	Proposed
Cameraman	31.91	31.81	32.18	32.17	32.15	**33.01**
	0.72	0.71	0.76	0.80	0.79	**0.86**
	0.39	0.33	0.39	0.44	0.47	**0.59**
	0.51	29.86	0.88	136.75	172.14	3.51
Barbara	33.69	31.81	32.15	43.26	34.27	**34.15**
	0.77	0.75	0.77	0.83	**0.87**	0.86
	0.41	0.40	0.43	0.50	0.59	0.61
	3.05	178.25	**1.64**	825.50	986.36	7.62
Lena	34.94	32.91	33.44	33.54	33.53	**33.94**
	0.70	0.77	0.78	0.82	**0.88**	0.86
	0.40	0.41	0.43	0.48	0.55	**0.59**
	0.46	30.31	0.90	134.41	165.01	4.90
House	34.94	34.14	34.55	35.15	35.14	**35.17**
	0.69	0.78	0.81	0.81	0.83	**0.87**
	0.44	0.43	0.51	0.58	0.58	**0.60**
	0.78	32.32	0.87	136.16	170.27	4.74
Peppers	32.70	32.58	33.03	32.97	32.97	**33.01**
	0.66	0.81	0.83	0.84	**0.85**	0.84
	0.32	0.35	0.42	0.55	**0.58**	**0.58**
	0.46	29.55	0.91	132.65	160.62	3.89
Baboon	27.78	27.85	27.89	27.93	27.89	**28.18**
	0.64	0.77	0.79	0.80	**0.84**	**0.84**
	0.28	0.28	0.44	0.50	0.53	**0.56**
	1.49	111.98	**1.22**	490.38	654.31	7.11

Table 3: The results of the different method for standard Gaussian white noise with $\sigma = 25$.

PSNR SSIM EPI Time/s	BM3D	EPLL	TNRD	WNNM	WSNM	Proposed
Cameraman	29.45	29.23	29.71	29.63	29.70	30.12
	0.70	0.70	0.74	0.71	0.75	0.76
	0.33	0.31	0.35	0.45	0.45	0.50
	0.54	31.43	0.92	261.72	289.96	4.57
Barbara	31.18	28.98	29.63	31.83	31.84	31.27
	0.75	0.71	0.75	0.77	0.82	0.85
	0.37	0.33	0.40	0.44	0.56	0.60
	3.14	179.30	1.64	1706.84	2350.22	10.62
Lena	30.60	30.30	30.77	30.88	30.89	32.27
	0.68	0.72	0.75	0.80	0.80	0.82
	0.32	0.36	0.40	0.46	0.47	0.48
	0.47	34.78	0.91	259.41	287.63	6.22
House	32.86	32.04	32.54	33.23	33.23	33.28
	0.66	0.76	0.80	0.83	0.80	0.83
	0.29	0.40	0.50	0.56	0.56	0.57
	1.49	31.16	0.92	263.04	298.06	6.43
Peppers	30.16	30.07	30.55	30.40	30.41	31.67
	0.64	0.77	0.81	0.81	0.82	0.80
	0.26	0.31	0.40	0.40	0.44	0.55
	0.44	32.25	0.87	259.73	291.05	4.88
Baboon	25.23	25.34	25.39	25.42	25.44	26.16
	0.62	0.71	0.77	0.77	0.78	0.80
	0.26	0.25	0.34	0.46	0.46	0.50
	1.62	106.28	1.23	986.96	1296.27	9.60

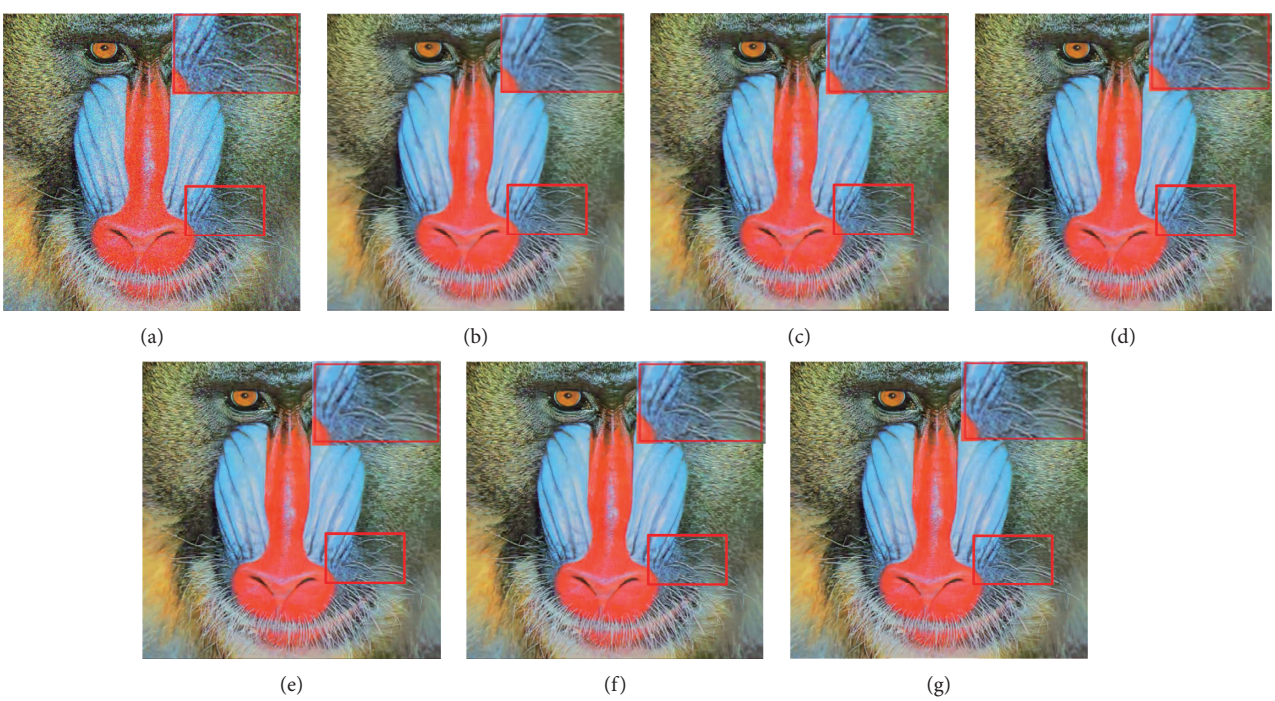


Figure 8: The denoising results of different methods with the noise level $\sigma = 25$. (a) Noise image (20.17 db), (b) BM3D (25.23 db), (c) EPLL (25.34 db), (d) TNRD (25.39 db), (e) TNNM (25.42 db), (f) WSNM (25.44 db), and (g) proposed (26.16 db).

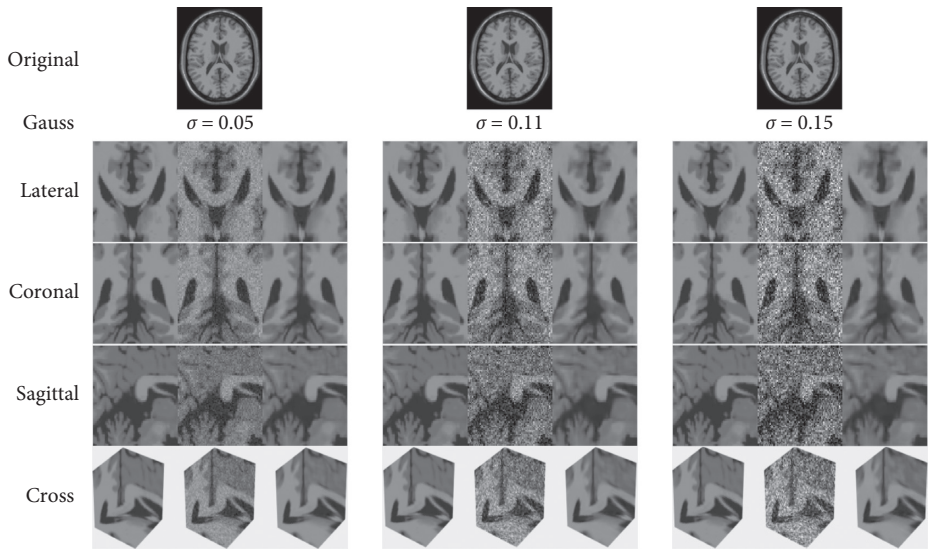


Figure 9: The results of removing the Gauss noise: from the left to right: the original image, the noisy image, and the denoising result.

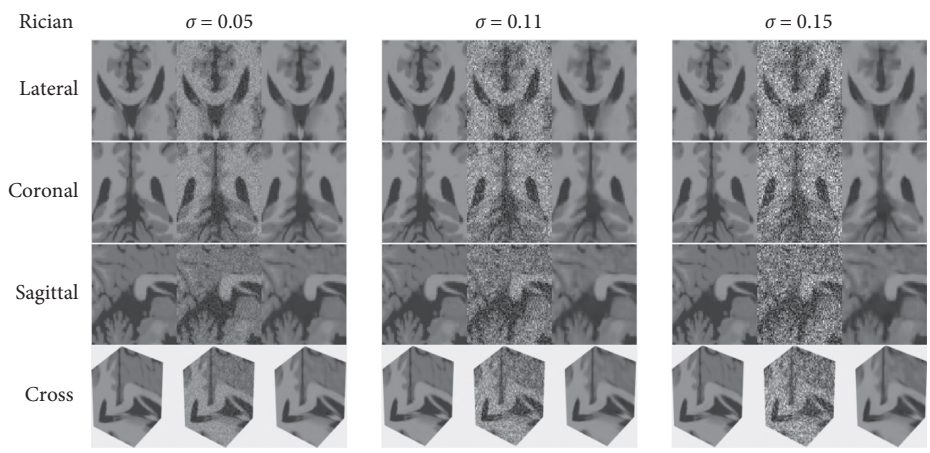


Figure 10: The results of removing the Rician noise: from the left to right: the original image, the noisy image, and the denoising result.

Based on the edge contrast, the edge-preserving index (EPI) is defined as

$$EPI = \frac{\left[\sum\left|p_s(i,j) - p_s(i+1,j)\right| + \left|p_s(i,j) - p_s(i,j+1)\right|\right]}{\left[\sum\left|p_o(i,j) - p_o(i+1,j)\right| + \left|p_o(i,j) - p_o(i,j+1)\right|\right]}, \tag{20}$$

where $p_s(i,j)$ is a pixel from the testing image, $p_o(i,j)$ is a pixel from the original image, p_s and p_o are located in the edge region, i is the number of rows, and j is the number of columns. The range of EPI is 0 to 1. When EPI equals to 1, the edge of the image is completely maintained. When the EPI equals to 0, it means the image becomes a plane without any change. The larger the EPI value, the stronger the edge holding ability.

3.1. Experiment 1. Adding to the Gaussian white noise with different standard deviations, the first experiment is implemented under some popularly used natural images shown in Figure 7. Five famous methods are used to be the comparative methods, including BM3D [26], EPLL [12], TNRD [27], WNNM [28], and WSNM [29]and the proposed method (proposed for short). All the results are reported in Tables 1–3, respectively.

In addition, the visual performance of different methods when the noise level $\sigma = 25$ is shown in Figure 8 to provide the subjective comparison. A small region, marked by the red rectangle, is enlarged to clearly display the visual difference.

3.2. Experiment 2. The second experiment is implemented under Gaussian noise and Rician noise. The testing volume data is a T1 brain network voxel with a size of $181 \times 217 \times 181$. The slice thickness is 1 mm. Without loss of generality, it is assumed that all the images are normalized to a real-valued signal in the intensity range [0, 1]. The experimental results are shown in Figures 9 and 10.

In Figures 9 and 10, the first row is one slice of the original volume data. In each column, it shows the lateral,

Table 4: The results of the different methods for Gaussian noise.

PSNR SSIM EPI	Filter	σ/%									
		1	3	5	7	9	11	13	15	17	19
Gauss noise	Noisy data	40.00	30.46	26.02	23.10	20.91	19.17	17.72	16.48	15.39	14.42
		0.97	0.81	0.66	0.53	0.43	0.36	0.30	0.25	0.22	0.19
		0.20	0.19	0.17	0.17	0.14	0.13	0.10	0.09	0.06	0.04
	OB-NLM3D	42.47	37.57	34.73	32.82	31.42	30.32	29.40	28.61	27.91	27.28
		0.99	0.97	0.95	0.92	0.90	0.87	0.84	0.82	0.79	0.77
		0.29	0.28	0.26	0.25	0.23	0.20	0.19	0.17	0.15	0.14
	OB-NLM3D-WM	42.52	37.75	35.01	33.13	31.73	30.61	29.68	28.88	28.18	27.55
		0.99	0.97	0.95	0.93	0.90	0.88	0.85	0.83	0.80	0.78
		0.32	0.30	0.29	0.28	0.26	0.26	0.24	0.22	0.20	0.18
	ODCT3D	43.78	37.53	34.89	33.18	31.91	30.90	30.07	29.35	28.73	28.18
		0.99	0.97	0.95	0.93	0.91	0.89	0.88	0.86	0.85	0.83
		0.47	0.46	0.45	0.42	0.40	0.37	0.36	0.31	0.28	0.25
	PRI-NLM3D	44.04	38.26	35.51	33.67	32.37	31.29	30.40	29.65	28.99	28.40
		0.99	0.98	0.96	0.94	0.92	0.90	0.89	0.87	0.85	0.84
		0.66	0.64	0.62	0.60	0.59	0.56	0.55	0.55	0.53	0.51
	BM4D	44.09	38.39	35.95	34.38	33.21	32.28	31.50	30.82	30.23	29.70
		0.99	0.98	0.96	0.95	0.93	0.92	0.91	0.90	0.88	0.87
		0.70	0.69	0.68	0.67	0.60	0.58	0.56	0.56	0.55	0.54
	Proposed	45.08	39.31	36.79	35.12	33.79	33.31	32.16	31.58	30.79	30.53
		0.99	0.97	0.96	0.95	0.93	0.92	0.91	0.90	0.88	0.88
		0.75	0.72	0.68	0.67	0.63	0.61	0.60	0.59	0.57	0.55

Table 5: The results of the different methods for Rician noise.

PSNR SSIM EPI	Filter	σ/%									
		1	3	5	7	9	11	13	15	17	19
Rician noise	Noisy data	40.00	30.49	26.09	23.20	21.04	19.32	17.88	16.65	15.57	14.60
		0.97	0.81	0.66	0.53	0.43	0.36	0.30	0.25	0.21	0.18
		0.20	0.19	0.17	0.17	0.14	0.13	0.10	0.09	0.06	0.04
	OB-NLM3D	42.41	37.45	34.54	32.51	30.97	29.71	28.62	27.64	26.74	25.91
		0.99	0.97	0.94	0.91	0.88	0.85	0.81	0.78	0.74	0.70
		0.29	0.28	0.26	0.25	0.23	0.21	0.19	0.17	0.15	0.14
	OB-NLM3D-WM	42.44	37.54	34.66	32.61	31.01	29.69	28.53	27.50	26.57	25.71
		0.99	0.97	0.95	0.92	0.88	0.85	0.81	0.77	0.74	0.70
		0.32	0.30	0.29	0.28	0.27	0.27	0.26	0.23	0.23	0.20
	ODCT3D	42.96	37.38	34.70	32.90	31.53	30.41	29.48	28.67	27.95	27.30
		0.99	0.97	0.95	0.93	0.90	0.88	0.86	0.84	0.82	0.80
		0.47	0.47	0.45	0.44	0.42	0.39	0.37	0.35	0.34	0.33
	PRI-NLM3D	43.97	38.19	35.34	33.37	31.94	30.74	29.75	28.88	28.10	27.39
		0.99	0.98	0.96	0.94	0.91	0.89	0.87	0.85	0.82	0.80
		0.66	0.65	0.63	0.60	0.57	0.55	0.54	0.52	0.48	0.46
	BM4D	44.08	38.34	35.83	34.17	32.89	31.82	30.90	30.06	29.29	28.57
		0.99	0.98	0.96	0.94	0.93	0.91	0.89	0.88	0.86	0.84
		0.71	0.70	0.69	0.67	0.62	0.60	0.59	0.57	0.56	0.55
	Proposed	45.10	39.49	36.97	35.38	33.61	32.92	32.18	31.61	30.42	29.71
		0.99	0.97	0.96	0.94	0.93	0.91	0.90	0.89	0.87	0.86
		0.76	0.71	0.68	0.67	0.64	0.63	0.62	0.60	0.58	0.57

coronal, sagittal, and cross results when the standard deviation is 0.05, 0.11, and 0.015, respectively. The results of under OB-NLM3D [30], OB-NLM3D-WM [31], ODCT3D [32], PRI-NLM3D [33], BM4D [34] and the proposed method are reported in Tables 4 and 5. All the quantitative results are shown in Tables 4 and 5, respectively.

By the comparison of all the above experimental results, it strongly demonstrates the efficiency of the proposed fusion methods, both visually and quantitatively. Specially, among the algorithms considered, the PSNR and SSIM performance of the proposed method shows that the better results can be obtained, when the noise level increases. In addition, better visual effects can be obtained from the smoothness of flat areas, the preservation of details along the edges, and the accurate restoration of phantom intensity.

The time complexity of the proposed method is $O(N \log_2 N)$. The cost of computing time is indeed an annoying problem at present. The main reason is that the process of our method contains more steps than other methods. The good news is that our method is not the

slowest. We think it will be effectively solved in the future by improving the hardware conditions. In addition, some parallel computing methods (CUDA will be used in our plan) can also be used to reduce the time cost. This is also the work we will deal in the future.

4. Conclusion

In this paper, an image denoising method that is based on the BM4D in the 3D shearlet transform domain and GAN is proposed. The proposed method can make full use of the sparse representation of the 3D shearlet transform and the good deeply learning ability of the generative adversarial network. All the experimental results prove the effectiveness and accuracy of the proposed method in the form of subjective comparison and objective quantification, strongly demonstrating the superiority of the proposed method over the traditional image denoising methods.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by a Project of Shandong Province Higher Educational Science and Technology Program (J18KA362), the National Natural Science Foundation of China (61502282, 61902222), the Natural Science Foundation of Shandong Province (ZR2015FQ005), the Taishan Scholars Program of Shandong Province (tsqn201909109), and the Open Fund Project of Shandong Key Laboratory of Information Technology of Intelligent Mine in Shandong University of Science and Technology.

References

[1] T. Yongpeng, J. Yu, and X. Cong, "Image denoising algorithm based on grouped dictionary and variational model," *Computer Application*, vol. 39, no. 2, pp. 551–555, 2019.

[2] S. Yongkui, T. Wen, and C. Tongwen, "A method to remove chattering alarms using median filters," *ISA Transactions*, vol. 73, no. 1, pp. 201–207, 2018.

[3] L. Zhiliang, S. Jia, G. Lihui et al., "Experimental study on active noise control based on wiener filtering," *Technology Wind*, vol. 5, no. 15, pp. 216-217, 2018.

[4] M. Xiaole, H. Shaohai, and Y. Dongsheng, "SAR image denoising based on residual image fusion and sparse representation," *KSII Transactions on Internet & Information Systems*, vol. 13, no. 7, pp. 3620–3637, 2019.

[5] H. Hao and L. Wu, "A new complex valued dictionary learning method for group-sparse representation," *Optik*, vol. 196, p. 163150, 2019.

[6] L. Shuaiqi, L. Ming, L. Peifei et al., "SAR image denoising via sparse representation in shearlet domain based on continuous cycle spinning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 5, pp. 2985–2992, 2017.

[7] C. Bo, D. Ning, and B. Jing, "Application of BM3D denoising algorithms in instrument image recognition," *Zhejiang Electric Power*, vol. 38, no. 3, pp. 54–58, 2019.

[8] Z. Wenwen and H. Yusheng, "Low rank sparse image denoising based on non-local self-similarity," *Computer Application*, vol. 38, no. 9, pp. 2696–2700+2746, 2018.

[9] X. Su and Z. Yingyue, "Non-local mean denoising based on image segmentation," *Computer Application*, vol. 37, no. 7, pp. 2078–2083, 2017.

[10] L. Yu and C. Sheng, "Summary of medical image segmentation methods," *Electronic Technology*, vol. 30, no. 8, pp. 169–172, 2017.

[11] W. Rui and Z. Youchun, "Wavelet threshold denoising under new threshold function," *Computer Engineering and Application*, vol. 49, no. 15, pp. 215–218, 2013.

[12] X. Wang, Q. Tao, L. Wang et al., "Deep convolutional architecture for natural image denoising," *International Conference Wireless Communications and Signal Processing*, vol. 5, no. 53, pp. 1–4, 2015.

[13] Y. Min, W. Xintong, L. Yao et al., "Plausibility-promoting generative adversarial network for abstractive text summarization with multi-task constraint," *Information Sciences*, vol. 521, pp. 46–61, 2020.

[14] P. Siddharth, R. Pranshu, and T. Jing, "An image augmentation approach using two-stage generative adversarial network for nuclei image segmentation," *Biomedical Signal Processing and Control*, vol. 57, p. 101782, 2020.

[15] M. Jiayi, Y. Wei, L. Pengwei et al., "FusionGAN: a generative adversarial network for infrared and visible image fusion," *Information Fusion*, vol. 48, pp. 11–26, 2019.

[16] B. Lei, Z. Xiongwei, Z. Yunfei et al., "Video saliency detection using 3D shearlet transform," *Multimedia Tools & Applications*, vol. 75, no. 13, pp. 7761–7778, 2016.

[17] D. Labate and P. Negi, "3D discrete shearlet transform and video denoising," *Proceedings of Spie the International Society for Optical Engineering*, vol. 8138, no. 3, pp. 815–822, 2011.

[18] D. Lijuan, W. Chunli, E. Qing et al., "Super-resolution algorithm of depth residual network image based on wavelet domain," *Journal of Software*, vol. 30, no. 4, pp. 941–953, 2019.

[19] L. Sun, B. Jenbo, Y. Zheng et al., "A novel weighted cross total variation method for hyperspectral image mixed denoising," *IEEE Access*, vol. 6, no. 1, pp. 172–188, 2017.

[20] C. Zailiang, Z. Ziyang, S. Hailan et al., "DN-GAN: Denoising generative adversarial networks for speckle noise reduction in optical coherence tomography images," *Biomedical Signal Processing and Control*, vol. 55, p. 101632, 2020.

[21] Z. Kai, Z. Wangmeng, C. Yunjin et al., "Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[22] L. Zhiyu, Z. Chengkun, and H. Min, "A nonsubsampled countourlet transform based CNN for real image denoising," *Signal Processing: Image Communication*, vol. 82, p. 115727, 2020.

[23] F. Fan, Y. Ma, C. Li, X. Mei, J. Huang, and J. Ma, "Hyperspectral image denoising with superpixel segmentation and low-rank representation," *Information Sciences*, vol. 397-398, pp. 48–68, 2017.

[24] M. Hongqiang, M. Shiping, X. Yuelei et al., "Image denoising based on improved stack sparse denoising self-encoder," *Computer Engineering and Application*, vol. 54, no. 4, pp. 199–204, 2018.

[25] H. R. Shahdoosti and Z. Rahemi, "Edge-preserving image denoising using a deep convolutional neural network," *Signal Processing*, vol. 159, pp. 20–32, 2019.

[26] X. Jia, Z. Junhua, and M. Lihui, "Denoising of salt and pepper noise based on improved BM3D algorithm," *Computer Engineering and Application*, vol. 54, no. 21, pp. 170–175, 2018.

[27] H. Burger, C. Schuler, and S. Harmeling, *Learning How to Combine Internal and External Denoising Methods. Pattern Recognition*, Springer, Berlin, Germany, pp. 121–130, 2013.

[28] G. Shuhang, Z. Lei, Z. Wangmeng et al., "Weighted nuclear norm minimization with application to image denoising," *Computer Vision and Pattern Recognition*, vol. 3, pp. 2862–2869, 2014.

[29] X. Yuan, G. Shuhang, L. Yan et al., "Weighted schatten p-norm minimization for image denoising and background subtraction," *IEEE Trans. Image Process.*vol. 8, no. 10, pp. 4842–4857, 2016.

[30] A. Welinton, S. Tiago, B. Gabriel et al., "Improving non-local video denoising with local binary patterns and image quantization," in *Proceedings of the 2016 29th SIBGRAPI Conference on Graphics*, pp. 1–9, Sao Paulo, Brazil, October 2016.

[31] P. Coupé, P. Hellier, S. Prima, C. Kervrann, and C. Barillot, "3D wavelet subbands mixing for image denoising," *International Journal of Biomedical Imaging*, vol. 2008, Article ID 590183, 11 pages, 2008.

[32] L. Xiangbo and Q. Tianshuang, "Denoise MRI images using sparse 3D transformation domain collaborative filtering," in *Proceedings of the 2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)*, vol. 1, pp. 233–236, Shanghai, China, October 2011.

[33] Y. Daitianyi, "Application of PRI_NLM3D denoising algorithms in CBCT 3D images," *New National Radiographic Digital Imaging and CT Technology*, vol. 10, no. 8, pp. 99–108, 2012.

[34] X. Ping, C. Bingqiang, L. Xue et al., "A new MNF-BM4D denoising algorithm based on guided filtering for hyperspectral images," *ISA Transactions*, vol. 2, no. 3, pp. 1–10, 2019.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1291526, 14 pages
https://doi.org/10.1155/2020/1291526

Research Article

Adaptive Cat Swarm Optimization Algorithm and Its Applications in Vehicle Routing Problems

Xiao-Fang Ji,[1] Jeng-Shyang Pan ,[1,2] Shu-Chuan Chu,[1,3] Pei Hu,[1,4] Qing-Wei Chai,[1] and Ping Zhang[5]

[1]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2]*College of Information Science Technology, Dalian Maritime University, Dalian 116026, China*
[3]*College of Science and Engineering, Flinders University, Sturt Rd, Bedford Park SA 5042, Adelaide, Australia*
[4]*School of Software, Nanyang Institute of Technology, Nanyang 473004, China*
[5]*Fuzhou Survey Institute, Fuzhou 350108, China*

Correspondence should be addressed to Jeng-Shyang Pan; jengshyangpan@gmail.com

Received 28 January 2020; Accepted 18 February 2020; Published 21 April 2020

Guest Editor: Chi-Hua Chen

This paper proposes a novel hybrid algorithm named Adaptive Cat Swarm Optimization (ACSO). It combines the benefits of two swarm intelligence algorithms, CSO and APSO, and presents better search results. Firstly, some strategies are implemented to improve the performance of the proposed hybrid algorithm. The tracing *radius* of the cat group is limited, and the random number parameter r is adaptive adjusted. In addition, a scaling factor update method, called a memory factor y, is introduced into the proposed algorithm. They can be learnt very well so as to jump out of local optimums and speed up the global convergence. Secondly, by comparing the proposed algorithm with PSO, APSO, and CSO, 23 benchmark functions are verified by simulation experiments, which consists of unimodal, multimodal, and fixed-dimension multimodal. The results show the effectiveness and efficiency of the innovative hybrid algorithm. Lastly, the proposed ACSO is utilized to solve the Vehicle Routing Problem (VRP). Experimental findings also reveal the practicability of the ACSO through a comparison with certain existing methods.

1. Introduction

In recent decades, Evolutionary Computation (EC) has become a very popular research topic, and great progress has been made in both theory and practice. Swarm Intelligence is a part of Evolutionary Computation, which is also one of the research hotspots in the field of evolutionary computing. Many metaheuristic algorithms have been proposed, including but not limited to Differential Evolution (DE) [1–3] Algorithm, Genetic Algorithm (GA) [4, 5], Particle Swarm Optimization (PSO) [6–8], Cat Swarm Optimization (CSO) [9–11], Ant Colony Optimization (ACO) [12–15], and Bat Algorithm (BA) [16, 17]. Some intelligent computing technologies show great promise in many practical application scenarios, for example, wind power [18, 19], vehicle routing problem [20–23], and wireless sensors [24–31]. All above have been successfully in applied evolutionary calculations to improve their performance, and it is also one of the effective methods to solve traffic problems. However, the diversity of algorithms is affected by the "no free lunch" theorem, which has inspired people to propose more valuable algorithms.

PSO is considered to have the following advantages: few control times, easy to implement, and convenient to use. However, it is simple to fall into a local maximum stagnation in terms of convergence and search earlier than anticipated. Therefore, avoiding local optimal solutions and accelerating the rate of convergence are two important issues for intelligent algorithms. Later, many variants of PSO were derived. One of them is proposed by Zhan et al. [32], which sets four states according to the distance between particles, and adaptive adjusts parameters.

CSO has two submodes which are only suitable for small-scale population optimization. When the population size increases, the convergence rate will be slower. In order

to alleviate the previously mentioned disadvantages, it is a crucial balance between population diversity and convergence speed. Through the improved cat swarm algorithm, the parameter adaptability of ACSO can increase the diversity and flexibility of the population.

ACSO is an optimization algorithm which aims to improve their convergence and search capabilities for cat swarm and particle swarm algorithms. According to the experimental results, based on the benefits and disadvantages of the APSO and the CSO, ACSO has improved. The CSO is applicable to smaller populations. Due to the evolutionary state strategy, the APSO is suitable to avoid getting trapped in a local optimum. Compared with other evolutionary algorithms, the ACSO shows unparalleled advantages.

The paper's structure is detailed below. Section 2 introduces related research works: PSO, APSO, and CSO. ACSO's adaptive parameter settings are discussed in Section 3. Then, the performances of ACSO, PSO, APSO, and CSO algorithms are verified by using typical 23 benchmark functions. This is shown in Section 4 while applying the algorithm to VRP as described in Section 5. Finally, Section 6 summarizes the work of this paper, and the suggestions are described for future work.

2. Related Research Works

In this section, the basic theory of many traditional algorithms will be reviewed briefly, namely, Cat Swarm Optimization (CSO) and Particle Swarm Optimization (PSO) and the deformation of PSO, named Adaptive Particle Swarm Optimization (APSO). APSO has greatly improved convergence speed and search capabilities. By reviewing the APSO and CSO algorithms, we wondered if we could combine the two algorithms to upgrade CSO.

2.1. Particle Swarm Optimization. Originally devised in 1995 by Kennedy and Eberhart [6, 33], PSO is inspired by the behaviour socially of swarms of fish and flocks of birds. It has been widely concerned by people and has good development prospects. In the PSO algorithm, the solution in each optimization space is considered as "particles" with neither volume nor mass. All the particles are based on their own cognitive learning, solo flight experience, and companion flight experience. Therefore, when particles seek the best position in the optimized space, their flights are adjusted through information exchange.

Population is randomly initialized at first, and then each particle follows the current individual local optimum and the group global optimum to succeed in finding the optimal solution space. While the program is running, each particle implies a point in the decision space and contains two basic information, position and velocity. Particles update velocity and displacement through the following state transition equations:

$$\begin{aligned} v_i^d(t+1) &= \omega \cdot v_i^d(t) + c_1 \cdot r_1\left(p_{\text{best}}^d(t) - x_i^d(t)\right) \\ &\quad + c_2 \cdot r_2\left(g_{\text{best}}^d(t) - x_i^d(t)\right), \\ x_i^d(t+1) &= x_i^d(t) + v_i^d(t+1). \end{aligned} \tag{1}$$

The primary PSO is effortless with simply a few adjusted parameters: $v_i^d(t)$ indicates the velocity value of the ith particle in the D-dimension before update; $x_i^d(t)$ denotes the position of the ith particle before the update; $p_{\text{best}}^d(t)$ represents the position of the particle in the particle swarm that currently has a locally optimal solution; $g_{\text{best}}^d(t)$ is the global optimal position of the ith particle; ω is the inertial weight of the particle; c_1 and c_2 also called the acceleration coefficients, for extending the velocity of the cat to move in the solution space and usually set to 2.05; r_1 and r_2 are two uniform random values uniformly generated in the range of [0, 1].

2.2. Adaptive Particle Swarm Optimization. Traditional PSO still has some shortcomings in global search and convergence. The algorithm has attracted the strong interest of many scholars and is committed to improving the performance of the algorithm. APSO, which is suggested by Zhan et al. [32], accelerates the convergence speed even more. The population distribution state of the algorithm includes four evolutionary states, namely, detection, development, convergence, and bounce. In the process of operation, the inertia weight, acceleration factor, and other parameters can be automatically controlled. Therefore, the search efficiency and the convergence speed are effectively improved. As the evolution progresses, particles may cluster together and converge to a local optimum. At this time, an elite learning strategy guides the global optimal particles to escape from local optimum. This algorithm breaks through the PSO, by detecting the distribution information of different populations and using this information to evaluate the evolutionary state, and the steps are shown below:

Step 1: the distribution information can use Euclidean distance to describe the average distance between each particle i and other particles:

$$\text{dist} = \frac{1}{n-1} \sum_{j=1, j\neq i}^{n} \sqrt{\sum_{k=1}^{d} \left(x_i^k - x_j^k\right)^2}, \tag{2}$$

where n and d, respectively, represent the population size and dimension.

Step 2: compare the distances of globally optimal particles d_g to other particles and calculate the maximum $d_{\max}$ and minimum $d_{\min}$ distances. The "evolutionary factor" f is denoted as follows:

$$f = \frac{d_g - d_{\min}}{d_{\max} - d_{\min}} \in [0, 1]. \tag{3}$$

In the exploration phase, the f value is large; in the exploitation phase, the f value decreases rapidly; after the environment changes, it will reach the convergence phase; as the number of iterations continues to increase, the particles will jump out, causing the f value to become larger. The cycle then repeats itself.

Step 3: the four states S1, S2, S3, and S4 are divided by f. With regard to the order, they represent the circumstances of exploration, exploitation, convergence, and jumping-out, respectively. Generally, a larger adaptive weight ω

value is set in exploration, and a smaller value is set in exploitation. However, the evolution factor f also shares some characteristics of the inertial weight ω, so ω can follow f change. Four strategies are summarized in Table 1.

At the same time, when the denominator is greater than 3.0, the values of c_1 and c_2 are standardized:

$$c_i = \frac{c_i}{c_1 + c_2} 3.0, \quad i = 1, 2. \tag{4}$$

2.3. Cat Swarm Optimization. CSO is a heuristic global optimization method which was first presented by Taiwan scholar Chu et al. in 2006 [9]. CSO was proposed based on imitating the behaviour of cat. It has been analyzed that cats always spend most of their time to observe the surrounding environments first instead of hunting. Before the hunt, they alternate between moving slowly and staying at a location in a stationary state. This is called the seeking mode. Another submode is called the tracing mode. The CSO algorithm relies on the cooperation of these two states to obtain the optimal solution. Similar to PSO, every cat has its own velocity and position. MR defines how many cats in the overall cat group enter the seeking mode and how many cats enter the tracing mode. Flag is identifying which mode the cat is in. The optimal solution is a fitness value FS, representing the cat's accommodation to the function of fitness and the best position of the cat that has been obtained. The algorithm's specific stages are as follows:

(i) Initial population, each cat has D-dimensional coordinate values

(ii) Initialize the speed for randomizing the position of each dimension

(iii) According to the mixture ratio MR, the population is randomly divided into seeking and tracing modes

(iv) On the basis of the cat's flag bit, perform the corresponding position update on the cat

(v) Evaluate and record the fitness function value of each cat and keep the cat with the best fitness

(vi) Terminate the algorithm if the conditions are met; otherwise, return to step three

2.3.1. Seeking Model. The mode in which cats look around to find targets is called the seeking mode. There are four key parameters:

Seeking Memory Pool (SMP) defines the search memory size of each cat, which is representative of the position features that the cat has sought out.

Seeking Range of the Selected Dimension (SRD) represents the change rate of the selected area, and the change range of each dimension is determined by the SRD change domain.

Counts of Dimension to Change (CDC) refers to the number of dimensions that a single cat will mutate in the future. Its value is a random value between 0 and the maximum dimension.

Self-Position Consideration (SPC) is a Boolean valued variable, which indicates whether the position of the cat is about to move includes the position that has passed.

Table 1: Control strategies for c_1 and c_2 in four states.

State	Strategy	c_1	c_2
Exploration	Strategy 1	Increase	Decrease
Exploitation	Strategy 2	Increase slightly	Decrease slightly
Convergence	Strategy 3	Increase slightly	Increase slightly
Jumping-out	Strategy 4	Decrease	Increase

The process is described below:

(i) Make $j = \text{SMP}$, which represents copy the cat's current position. In case, the value of SPC is true, let $j = \text{SMP}-1$, and then return to the current position as a candidate solution.

(ii) For each individual copy in the memory pool, according to the size of the CDC, randomly add or subtract SRD percentage to the current value, and the old value is replaced by the updated value.

(iii) Calculate the values of the fitness function FS of all solutions that are candidates in the memory pool.

(iv) If all fitness function values FS are not completely equal, calculate the selection probability of each candidate solution by equation (5).

(v) The candidate point with the highest fitness value is selected from the memory pool to replace the current cat position, thereby completing the cat position update:

$$P_i = \frac{|FS_i - FS_b|}{FS_{\max} - FS_{\min}}, \quad \text{where } 0 < i < j. \tag{5}$$

If the fitness function finds the minimum solution of the problem, let $FS_b = FS_{\max}$; otherwise, $FS_b = FS_{\min}$.

2.3.2. Tracing Model. The state mode when tracing the target after finding it is called the tracing mode. This action can be briefly described in three steps.

The process is as follows:

(i) Update the velocities of each dimension according to (6), the best position that the entire cat group passes, that is, the optimal solution currently found:

$$v_i^d(t+1) = v_i^d(t) + r \cdot c \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), \quad d = 1, 2, 3, \ldots, m. \tag{6}$$

(ii) Check whether the velocities are within the maximum velocity range.

(iii) According to (7), update the cat's position:

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1). \tag{7}$$

3. Adaptive Cat Swarm Optimization

This paper proposes a new cat swarm algorithm with adaptive strategy based on the traditional APSO and CSO algorithms. This improvement does not only advance the efficiency and convergence of the algorithm but also maintains an understanding of the uniformity of the distribution. The specific content and innovations can be summarized as follows.

3.1. Increase Adaptive Parameters. Based on the parameter self-adaptation, the operation process of random numbers is adjust, and an adaptive strategy is add. In the early stage of iteration, the cat swarm can obtain strong global optimization ability:

$$r_k = \frac{1-\left(\sqrt{c_k}+1\right)\cdot rand\cdot i}{c_k\cdot \text{MaxIter}}, \quad k=1,2,\ i<\frac{\text{MaxIter}}{2}. \tag{8}$$

Make the particle swarm converge to the optimal position later in the iteration:

$$r_k = \frac{1-\left(\sqrt{c_k}+1\right)\cdot (1+rand)\cdot i}{c_k\cdot \text{MaxIter}}, \quad k=1,2,\ i\geq\frac{\text{MaxIter}}{2}. \tag{9}$$

When the initial values of c_1 and c_2 are set relatively small, add them to limit the generation of negative values, in order to avoid negative numbers:

$$r_1 = r_2 = rand, \quad r_1\leq 0,\ r_2\leq 0. \tag{10}$$

Change equations (8)–(10) are beneficial to the global search ability at the early stage of particle iteration, local refinement in later iterations, and to improve the accuracy of the solution.

3.2. A Radius Range is Added to the Search Position. When the distance between x_i^d and g_{best}^d is less than the radius, then toward the individual x_i^d to g_{best}^d. However, when the distance between x_i^d and p_{best}^d is less than the radius, just deviate it from p_{best}^d. The value of the elements is defined by the following equations:

$$\begin{aligned} v_i^d(t+1) &= \omega\cdot v_i^d(t) + c_1\cdot r_1\cdot\left(p_{\text{best}}^d(t)-x_i^d(t)\right) \\ &\quad + c_2\cdot r_2\cdot\left(g_{\text{best}}^d(t)-x_i^d(t)\right) + f\cdot F_i + e\cdot E_i, \end{aligned} \tag{11}$$

$$\begin{cases} f = 0.1 - i\cdot\left(\dfrac{0.2}{\text{MaxIter}}\right), \\ e = 2\cdot rand, \\ F_i = g_{\text{best}}^d(t) - x_i^d(t), \quad dist2gbest\leq radius, \\ E_i = p_{\text{best}}^d(t) + x_i^d(t), \quad dist2pbest\leq radius, \end{cases} \tag{12}$$

where f and e show the weights attracted toward the global optimal solution and the weights away from the local optimal solution, respectively. F_i and E_i indicate the food sources of the ith individual and the enemy of the ith individual.

3.3. Increase Memory Factor y. Particles learn from each other to obtain the most informative information in their respective fields. For each searcher, a memory factor y is added, and each particle gives a lower memory weight to the previous position. For the purpose of updating the historical best position of each searcher, higher memory weight is referenced to update the current position:

$$\begin{aligned} x_i^d(t+1) &= \frac{1}{2}\Big(y\cdot x_i^d(t) + (1-y)\cdot x_i^d(t-1) \\ &\quad + y\cdot v_i^d(t+1) + (1-y)\cdot v_i^d(t)\Big). \end{aligned} \tag{13}$$

To better explain the process of ACSO, the complete flow chart is shown in Figure 1. Firstly, randomly initialize each cat. Then, calculate the fitness value. Finally, make its parameters adaptive adjusted in the tracing mode.

The pseudocode of ACSO seeking mode function is exhibited in Algorithm 1.

The pseudocode of ACSO tracing mode function is exhibited in Algorithm 2.

4. Experiment and Result Analysis

This segment is predominantly to verify the efficient performance of the projected algorithm, and 23 mathematical optimization functions were performed for the purpose of comparing the ACSO with PSO, APSO, and CSO. These typical test equations are listed in Tables 2–4 used by many scholars [34]. Three of these elements need to be declared, *Space*, D_{im}, and $F_{\min}$, which denotes the boundary of function's search space, the dimension of the function, and the optimal solution, respectively.

4.1. Experimental Results. For verifying the results, CSO, APSO, and PSO are used to compare with the proposed ACSO algorithm. 23 benchmark functions are used to evaluate the performance of ACSO for real-parameter optimization. Usually, the benchmark function is also described as a mathematical test function. The mathematical test function used illustrates its 2D version in Figures 2–4. The relevant test parameters are listed in Table 5. In order to achieve a fair competition, for each test function, we tested 10 times for each optimization algorithm to get the average and standard deviation. The population size of each algorithm is set to 100, and the maximum number of iterations is 500. Subsequently, Table 6 illustrates the comparison of four algorithms on average (Ave) and standard deviation (Std).

4.2. Experiment Analysis. From Figures 5–7, solution quality and speed of PSO, APSO, CSO, and ACSO under 23 benchmark functions can be obtained. The horizontal axis in the figures stand for the maximum number of iterations during program execution and along the vertical axis are the corresponding fitness values. They are the simulation results

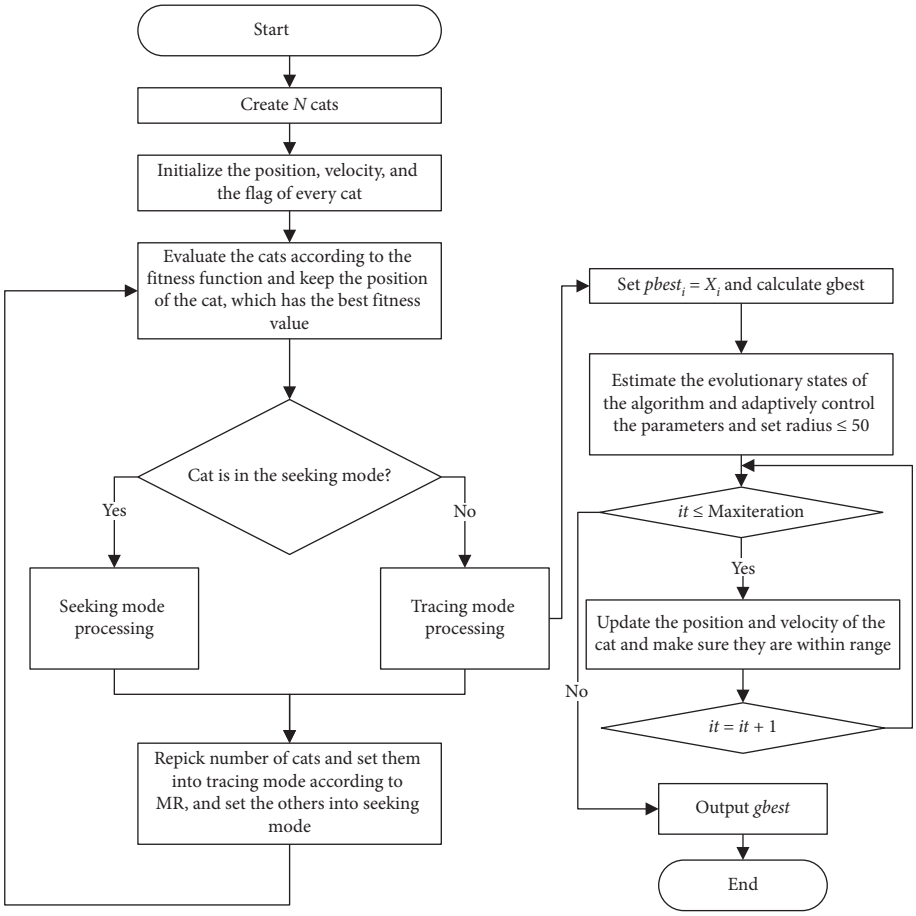


Figure 1: The complete flow chart of ACSO.

```
Create N cats
Initialize related parameters
Calculate the fitness of each cat
Divides cats into two mode based on flag
SMP = the search memory pool size
if flag = = 0 then
   for i = 1 to SMP do
      fitness = fobj(catCopy(i).Posi)
   end for
else
   for i = 2 to SMP do
      fitness = fobj(catCopy(i).Posi)
   end for
end if
```

Algorithm 1: ACSO seeking mode.

```
while t < = Maxiteration do
   Adaptive adjustment parameter r by
   equations (8)~(10)
   Compare the distance between Cat.Posi and
   cat's gbest.Posi and pbest.Posi
   Dist2gbest = distances (Cat.Posi, gbest.Posi)
   if Dist2gbest < = radius then
      F = gbest.Posi − Cat.Posi
   else
      F = 0
   end if
   Dist2pbest = distances (Cat.Posi, pbest.Posi)
   if Dist2pbest < = radius then
      E = pbest.Posi + Cat.Posi
   else
      E = 0
   end if
   Update the position and velocity by
   equations (11) and (12)
end while
return gbest
```

Algorithm 2: ACSO tracing mode.

of unimodal, multimodal, and fixed-dimensional multimodal mathematical test functions.

The unimodal function is a continuous function with only one extreme point in the domain; in other words, it has only one global optimum and no local optimum, so these algorithms are used to benchmark it. The performance of ACSO is superior to PSO, APSO, and ACSO. From Figure 5, because they contain only one global optimal solution, they have faster search convergence in the function.

The multimodal function is a function that contains multiple locally or globally optimal solutions, the purpose of which is to detect whether the test algorithm avoids local

Table 2: Unimodal benchmark functions.

Function	Space	D_{im}	$F_{\min}$
$F_1(x) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]$	30	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10, 10]$	30	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	$[-100, 100]$	30	0
$F_4(x) = \max i\{\lvert x_i \rvert, 1 \le i \le n\}$	$[-100, 100]$	30	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	30	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	$[-100, 100]$	30	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]$	30	0

Table 3: Multimodal benchmark functions.

Function	Space	D_{im}	$F_{\min}$
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500, 500]$	30	-12569
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	30	0
$F_{10}(x) = -20\exp(-0.2\sqrt{(1/n)\sum_{i=1}^{n} x_i^2})$ $-\exp((1/n)\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	$[-32, 32]$	30	0
$F_{11}(x) = (1/4000)\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(x_i/\sqrt{i}) + 1$	$[-600, 600]$	30	0
$F_{12}(x) = (\pi/n)\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$ $+\sum_{i=1}^{n} u(x_i, 10, 100, 4) y_i = 1 + ((x_i + 1)/4) u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]$	30	0
$F_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\}$ $+\sum_{i=1}^{n} u(x_i, 10, 100, 4)$	$[-50, 50]$	30	0

Table 4: Fixed-dimension multimodal benchmark functions.

Function	Space	D_{im}	$F_{\min}$
$F_{14}(x) = ((1/500)\sum_{j=1}^{25} 1/(j + \sum_{i=1}^{2}(x_i - a_{ij})^6))^{-1}$	$[-65, 65]$	2	1
$F_{15}(x) = \sum_{i=1}^{11} a_i - (x_1(b_i^2 + b_i x_2)/(b_i^2 + b_i x_3 + x_4))^2$	$[-5, 5]$	4	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + (1/3)x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]$	2	-1.0316
$F_{17}(x) = (x_2 - (5.1/4\pi^2)x_1^2 + (5/\pi)x_1 - 6)^2 + 10(1 - 1/8\pi)\cos x_1 + 10$	$[-5, 5]$	2	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	$[-2, 2]$	2	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	$[1, 3]$	3	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	$[0, 1]$	6	-3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)((X - a_i)^T)] + c_i]^{-1}$	$[0, 10]$	4	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)](X - a_i)^T + c_i^{-1}$	$[0, 10]$	4	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i^{-1}]$	$[0, 10]$	4	-10.5363

optimums. Relatively speaking, the improved algorithm is better than other algorithms, as shown in Figure 6. Premature stagnation of the optimal solution appears in F_9 and F_{11}. ACSO is better than other algorithms in the benchmark function of F_{10} and F_{13}. However, the PSO algorithm is known to have the shortcoming of premature convergence in dealing with multimodal optimization problems, owing to the lack of enough momentum for particles to do exploration or exploitation when the algorithm is nearing its end, so it is the worst solution curve.

In the fixed-dimensional multimodal function, Figure 7 shows the curves of simulation results by all four algorithms, where the curves of APSO, CSO, and ACSO are almost overlapped in F_{17}. From Table 6 and Figures 5–7, observing ACSO, not only can they avoid getting trapped in a local optimum but they can also converge quickly and eventually finding a global optimal solution. Therefore, we conclude that the proposed algorithms are noticeably superior to other comparison algorithms.

5. Adaptive Cat Swarm Algorithm Application in Vehicle Routing Problem

In this segment, the projected algorithm is connected to Vehicle Routing Problem (VRP). One of the fundamental problems in logistics has always been VRP, with the core link being cargo distribution. It was originally proposed by Dantzig and Ramser in 1959 [35]. Traditionally, VRP refers

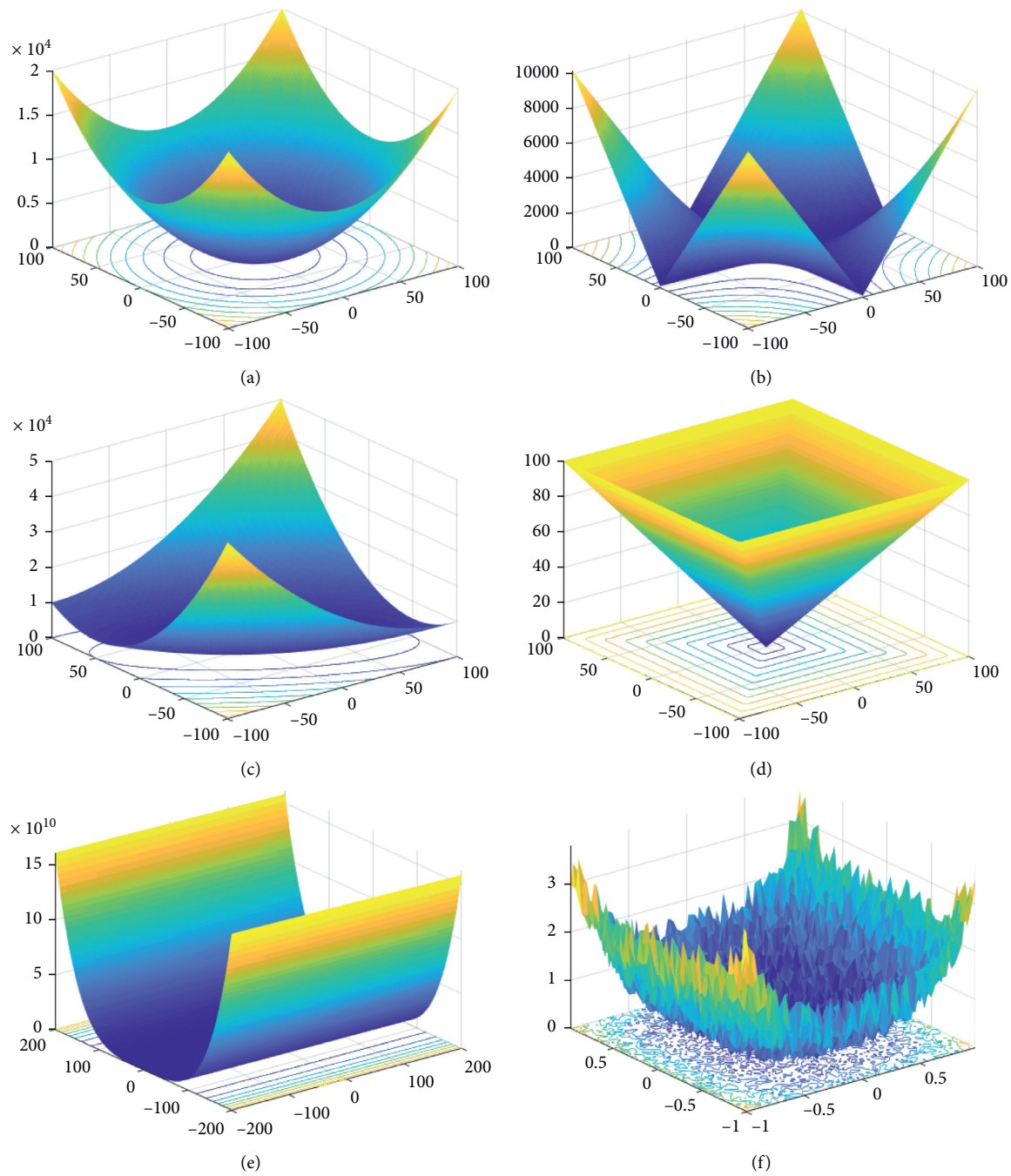


Figure 2: 2D versions of unimodal benchmark functions: (a) F1; (b) F2; (c) F3; (d) F4; (e) F5; (f) F7.

to the location of the distribution center that is known, the coordinate requirements and position of every customer, and finding the best route under precise constraints for visiting each customer, with the requirement of the lowest cost in transportation. The VRP has been studied for many years in the fields of mathematics and computer science. It also has the characteristics of nonlinearity, nonconvexity, complexity, and constraints and is difficult to coordinate with each other. In reality, its procedure of solution is quite complicated; therefore, there are certain advantages to solving the issue using an intelligent heuristic algorithm. Many scholars have devoted themselves to solving this problem and its derivative problems. The best known results for VRP have been obtained using Tabu Search (TS) [36, 37] or Simulated Annealing (SA) [38, 39]. In the present paper, VRP is chosen as the application objective of the ACSO algorithm to validate the algorithm's practicability further.

5.1. Description of Constraints

(i) The total cargo carried by each vehicle must meet its maximum load limit

(ii) Each vehicle may serve multiple customers, but each customer can only be served by one vehicle

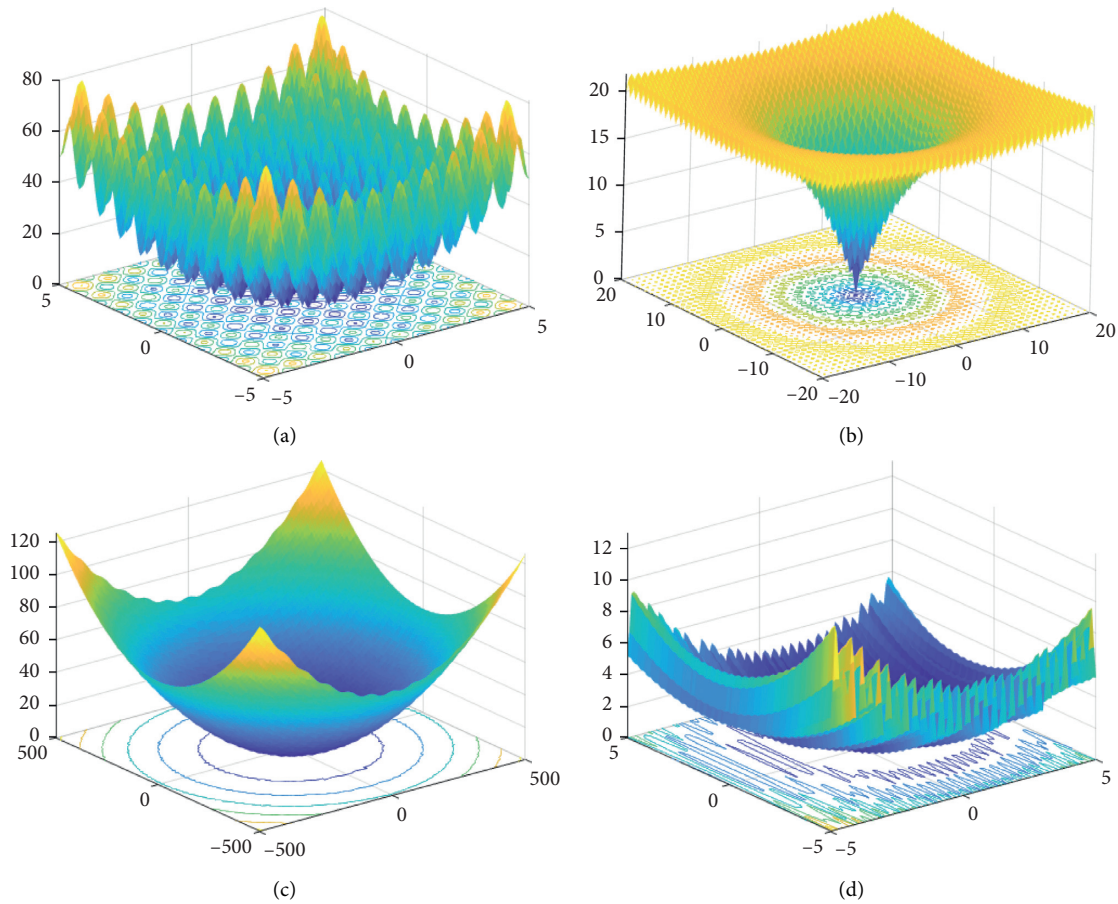


Figure 3: 2D versions of multimodal benchmark functions: (a) F9; (b) F10; (c) F11; (d) F13.

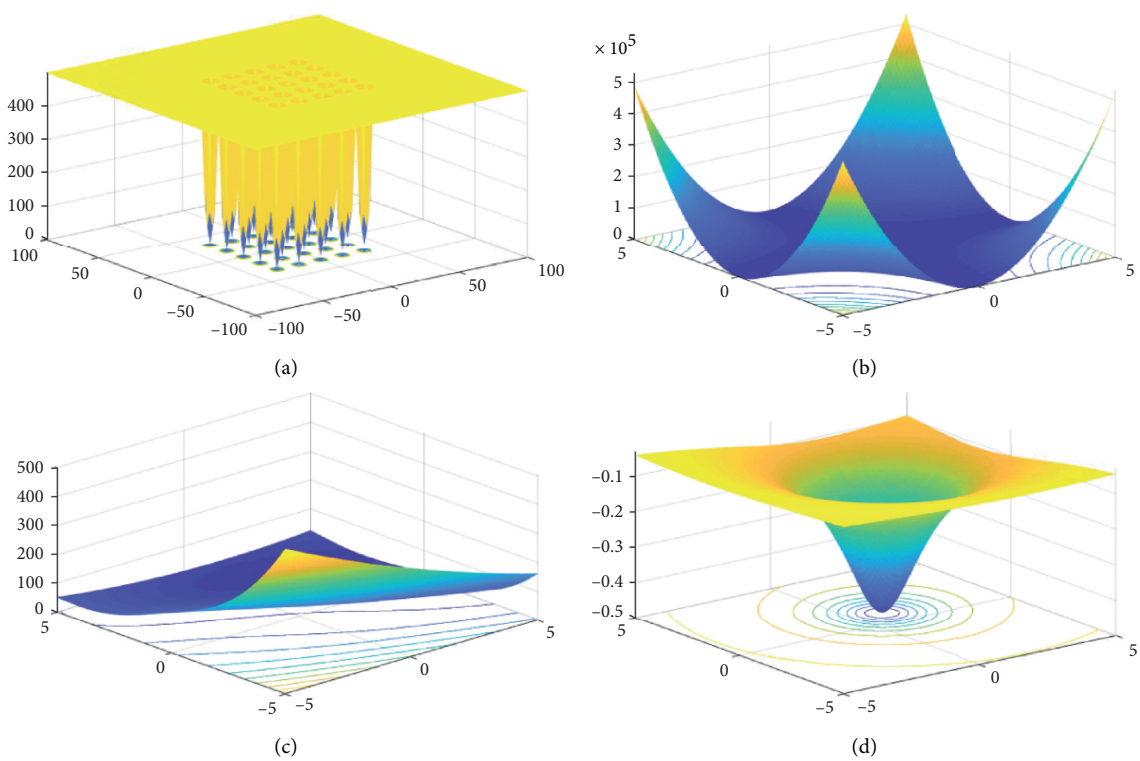


Figure 4: 2D versions of fixed-dimensional multimodal benchmark functions: (a) F14; (b) F15; (c) F17; (d) F21.

Table 5: Parameter setting of each algorithm.

Algorithm	Main parameter setting
PSO	VelMin = −6, VelMax = 6, $c = 2.5$, $w = 0.9$
APSO	VelMin = −6, VelMax = 6, $c1 = c2 = 2.5$, $w = 0.9$, status = "S1"
CSO	SRD = 0.9, MR = 0.3, RangeMin = −30, RangeMax = 30, VelMin = −6, VelMax = 6, c = 1.05, $w = 0.6$
ACSO	SRD = 0.9, MR = 0.3, RangeMin = −30, RangeMax = 30, VelMin = −6, VelMax = 6, $c1 = c2 = 1.05$, $w = 0.6$, status = "S1," radius = 50

Table 6: The statistical results of the algorithms.

Function	PSO		APSO		CSO		ACSO	
	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation
$F1$	4.66×10^{4}	3.94×10^{3}	1.46×10^{1}	7.28×10^{0}	2.43×10^{-33}	3.80×10^{-33}	1.62×10^{-48}	2.76×10^{-48}
$F2$	3.74×10^{33}	6.42×10^{33}	4.55×10^{1}	1.60×10^{1}	1.50×10^{-19}	2.59×10^{-19}	4.40×10^{-24}	6.61×10^{-24}
$F3$	7.92×10^{4}	2.85×10^{4}	5.29×10^{3}	3.68×10^{2}	6.36×10^{-27}	1.10×10^{-26}	45.04×10^{-46}	8.73×10^{-46}
$F4$	6.91×10^{1}	6.46×10^{0}	3.74×10^{0}	4.14×10^{-1}	2.92×10^{-16}	5.06×10^{-16}	1.33×10^{-25}	2.31×10^{-25}
$F5$	1.83×10^{10}	3.53×10^{9}	7.23×10^{3}	2.91×10^{3}	2.88×10^{1}	3.86×10^{-2}	2.84×10^{1}	1.70×10^{-5}
$F6$	3.89×10^{4}	1.03×10^{4}	2.59×10^{1}	1.52×10^{1}	2.51×10^{0}	1.14×10^{-1}	3.65×10^{0}	2.08×10^{-1}
$F7$	2.16×10^{9}	5.53×10^{8}	1.09×10^{3}	4.58×10^{2}	8.37×10^{-6}	6.77×10^{-6}	1.58×10^{-6}	1.71×10^{-6}
$F8$	-5.69×10^{2}	4.06×10^{1}	-1.17×10^{3}	8.78×10^{1}	1.22×10^{39}	2.12×10^{39}	-6.88×10^{2}	5.27×10^{2}
$F9$	4.91×10^{4}	2.63×10^{3}	3.19×10^{2}	4.51×10^{1}	0	0	0	0
$F10$	2.12×10^{1}	6.89×10^{-2}	1.77×10^{1}	4.12×10^{0}	8.88×10^{-16}	0	8.88×10^{-16}	0
$F11$	1.09×10^{1}	1.31×10^{0}	4.87×10^{-1}	1.09×10^{-1}	0	0	0	0
$F12$	5.97×10^{9}	2.74×10^{9}	5.88×10^{0}	3.74×10^{0}	2.04×10^{-1}	1.67×10^{-1}	4.24×10^{-1}	5.79×10^{-2}
$F13$	1.13×10^{10}	3.68×10^{9}	2.33×10^{1}	1.45×10^{1}	1.30×10^{0}	5.53×10^{-1}	2.68×10^{-4}	1.74×10^{-6}
$F14$	1.26×10^{1}	4.37×10^{0}	1.33×10^{0}	5.74×10^{-1}	4.33×10^{0}	2.33×10^{0}	1.12×10^{0}	1.38×10^{-1}
$F15$	1.79×10^{-2}	1.64×10^{-2}	1.01×10^{-3}	4.60×10^{-5}	8.14×10^{-4}	3.56×10^{-4}	3.51×10^{-4}	2.22×10^{-6}
$F16$	1.53×10^{4}	2.61×10^{4}	-1.03×10^{0}	0	-1.03×10^{1}	7.17×10^{-4}	-1.02×10^{0}	9.85×10^{-3}
$F17$	1.83×10^{0}	2.34×10^{0}	3.98×10^{-1}	0	4.02×10^{-1}	7.39×10^{-3}	3.98×10^{-1}	2.97×10^{-5}
$F18$	1.27×10^{8}	2.06×10^{8}	3.00	5.44×10^{-16}	3.00×10^{0}	8.84×10^{-4}	3.43×10^{0}	5.22×10^{-1}
$F19$	-2.07×10^{-33}	3.58×10^{-33}	-3.86×10^{0}	3.14×10^{-16}	-3.72×10^{0}	1.07×10^{-1}	-2.99×10^{0}	5.14×10^{-1}
$F20$	0	0	-6.59×10^{-1}	5.79×10^{-1}	-1.85×10^{0}	7.13×10^{-1}	-1.33×10^{0}	1.54×10^{-1}
$F21$	-3.59×10^{-2}	2.85×10^{-2}	-3.47×10^{0}	1.41×10^{0}	-2.14×10^{0}	8.35×10^{-1}	-1.02×10^{1}	6.80×10^{-9}
$F22$	-1.91×10^{-2}	1.40×10^{-2}	-1.04×10^{1}	1.78×10^{-15}	-3.66×10^{0}	1.04×10^{0}	-1.04×10^{1}	3.90×10^{-7}
$F23$	-1.10×10^{-2}	1.89×10^{-3}	-7.83×10^{0}	4.69×10^{0}	-3.00×10^{0}	7.05×10^{-1}	-1.05×10^{1}	1.38×10^{-7}

(iii) Vehicle k should go to the next customer i or returns to the distribution center immediately after serving customer j

5.2. Definition of Parameters. The vehicle routing problem is defined on a directed network $G = (V, E)$ with a vertex set $V = \{v_i \mid i = 0, 1, 2, 3, \ldots, n\}$, where v_0 represents a distribution center, $\{v_1, v_2, v_3, \ldots, v_n\}$ indicates the set of customers in a directed graph, $E = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\}$ is an edge set, and c_{ij} represents the attribute value of each edge.

(i) $N = \{1, 2, 3, \ldots, n\}$ is the collection of all customers

(ii) $K = \{1, 2, 3, \ldots, k\}$ is the collection of all delivery vehicles

(iii) c_0 is the unit distance cost

(iv) d_{ij} is the distance between the two points i and j

(v) c_{ij} is the transportation cost from point i to point j and $c_{ij} = c_0 d_{ij}$

(vi) r_i is the customer demands for goods

(vii) W is the maximum load capacity

(viii) S_k is the set of customer points for vehicle k service

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ depart from } i \text{ to } j,\ i, j \in N, \\ 0, & \text{otherwise,} \end{cases}$$
$$x_{ik} = \begin{cases} 1, & \text{if customer } i \text{ is served by vehicle } k, \quad i \in N, k \in K, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

The objectives and restrictions of the VRP are then described as follows:

$$\min \quad c = \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{K} c_{ij} x_{ijk}, \tag{15}$$

$$\sum_{i=0}^{N} \sum_{k=1}^{K} x_{ijk} = 1, \quad j \in N, \tag{16}$$

$$\sum_{i=1}^{N} \sum_{j=0}^{N} r_i x_{ijk} \leq W, \quad k \in K, \tag{17}$$

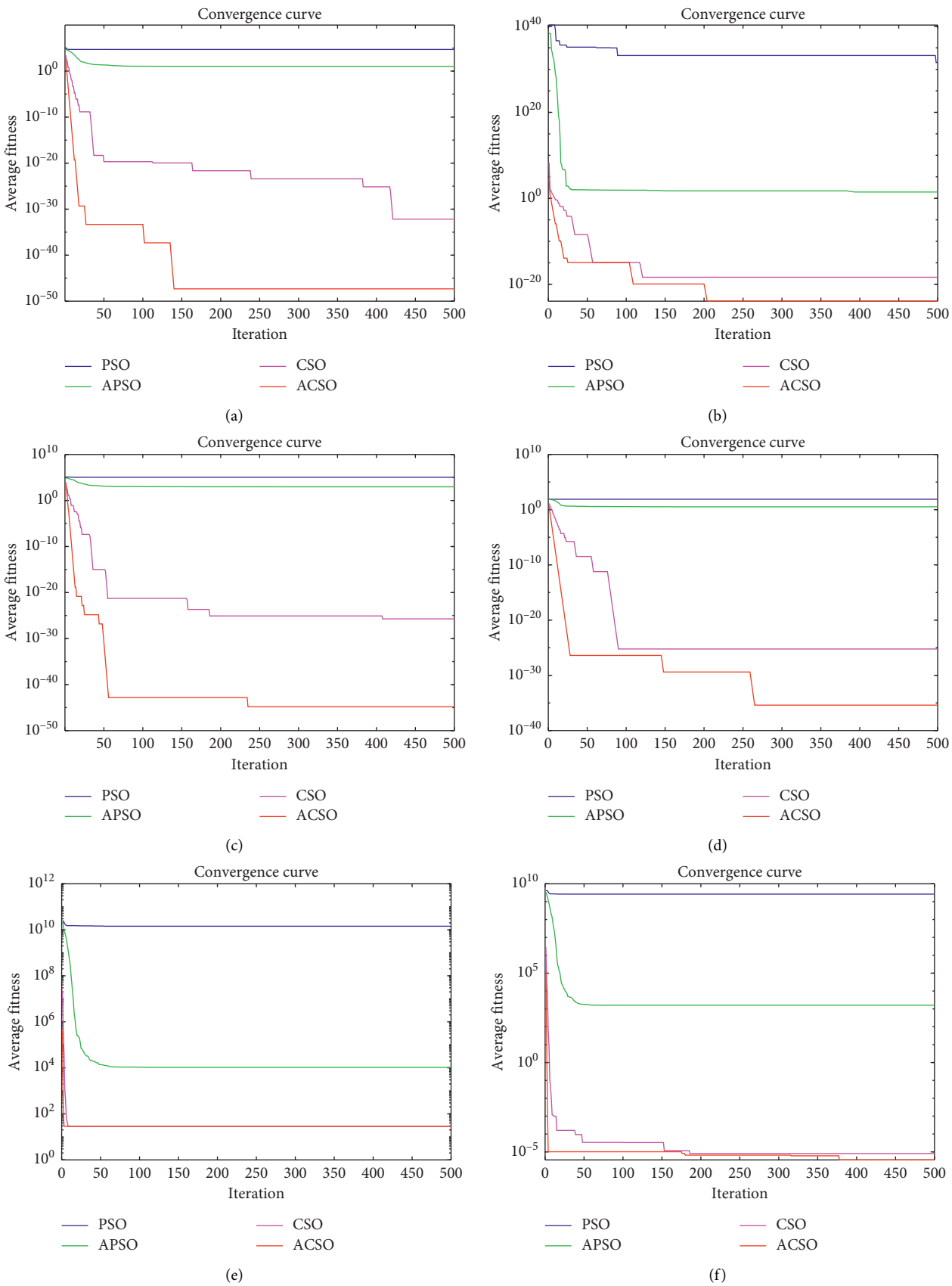


Figure 5: Simulation results for unimodal benchmark functions: (a) F1; (b) F2; (c) F3; (d) F4; (e) F5; (f) F7.

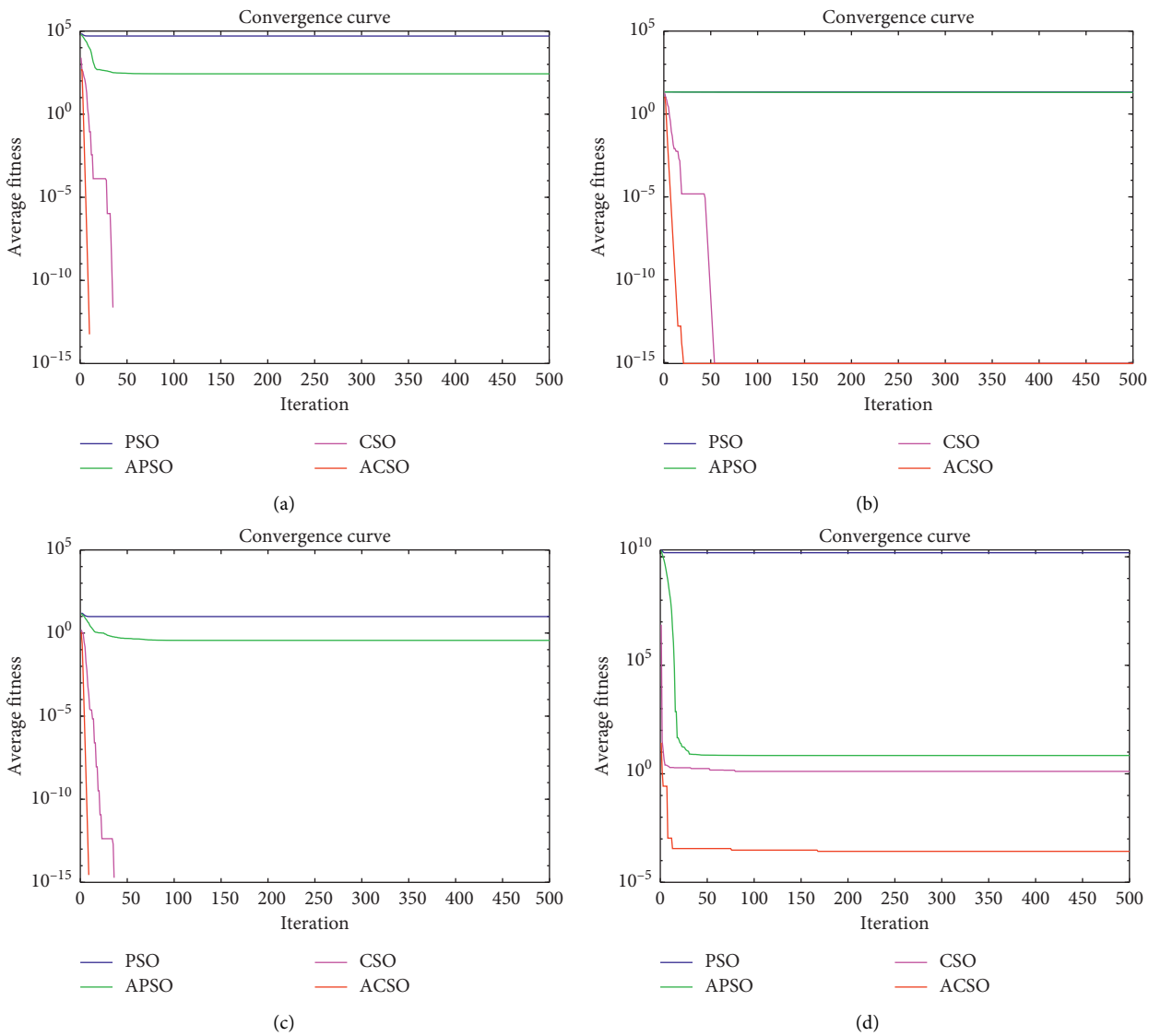


Figure 6: Simulation results for multimodal benchmark functions: (a) F9; (b) F10; (c) F11; (d) F13.

$$\sum_{i=1}^{N} x_{ijk} = \sum_{j=1}^{N} x_{jik}, \quad k \in K, \tag{18}$$

$$\sum_{i,j \in N} x_{ijk} = |S_k| - 1, \quad k \in K. \tag{19}$$

In VRP problems, as can be seen from the above model, equation (15) is a multimode function that belongs to the test function, aiming to minimize the objective function with least vehicle distribution cost. The level of distribution costs is the basic requirement for whether the economic benefits of the distribution process are maximized.

In a process with supply and demand in existence, formulas (16)–(19) denote the following: equation (16) demonstrates that this condition of constraint ensures a visit to the customer. Only one vehicle is allowed to supply each customer; equation (17) is a cargo flow constraint, requiring the load of the vehicle that cannot exceed the maximum load; equation (18) indicates that the continuity of the vehicle allocation process is constrained; that is, it must start from this point after serving customer i; equation (19) represents that each vehicle is restricted from having no subloops in the path.

5.3. Analysis of Experimental Results. In order to verify the effectiveness of ACSO algorithm to solve VRP, experiments were performed using MATLAB R2018b software. The simulation environment is the processor Inter (R) Core (TM) i7-8550U CPU @1.80 GHz 2.00 GHz, PC with Win10 operating system. The iteration is set as 300, and the statistical results of the algorithms used to solve VRP and the three algorithms of CSO, APSO, and PSO are recorded in Table 7.

The case column denotes various calculation examples, the Best Cost column represents the optimal value solution, and the Time/s column is representative of the whole time of the running of the algorithm. Through the comparison of the

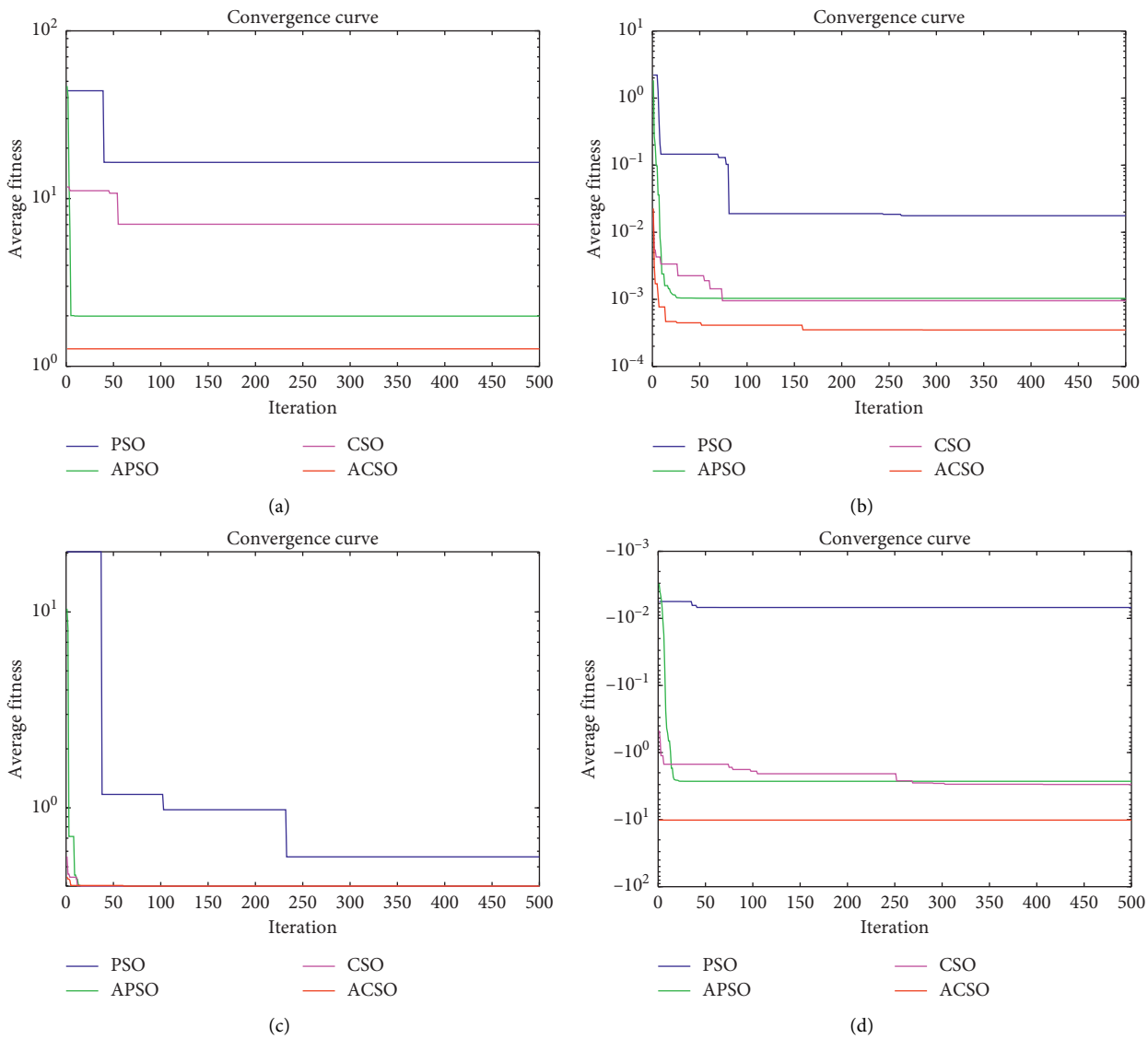


Figure 7: Simulation results for fixed-dimension multimodal benchmark functions: (a) F14; (b) F15; (c) F17; (d) F21.

Table 7: Comparison of optimal value and running time.

Case	ACSO		CSO		APSO		PSO	
	Best cost	Time (s)	Best cost	Time (s)	Best cost	Time (s)	Best cost	Time (s)
n14-k4	327.0238	105.9069	350.1766	91.4349	366.7308	93.6879	484.7995	93.8249
n20-k7	356.696	120.8684	374.5871	94.1607	383.2518	97.4590	554.6471	97.5232
n25-k5	489.5978	115.6802	527.2954	91.9872	543.6339	95.5831	695.325	95.8131
n30-k5	612.4174	121.0953	702.9065	94.8490	658.8413	100.1747	960.6692	93.3821
n40-k6	691.7631	137.0174	989.472	98.0159	917.9026	109.4481	1537.5096	94.3153

above experiments, it can be seen intuitively, compared with CSO, APSO, and PSO algorithms, the ACSO algorithm has achieved better results and has certain advantages. According to the results of this ACSO algorithm, however, as the number of nodes in the transmission network increases, it will lead to an increase in the time consumed during operation. To a certain extent, it shows that the algorithm still has room for improvement.

Table 8 presents the optimal consequence of ACSO according to the case n20-k7. Seven vehicles set off from the distribution center, each of them found an optimal path under specific constraints, traversing twenty customers to meet the customer's cargo needs. Here, zero point represents the distribution center, and numbers from one to twenty denote the customer number. Simultaneously, the optimal path graph is shown in Figure 8. "Star" in the figure

Table 8: Optimal consequence of ACSO.

Vehicle number	Delivery route
1	0-15-9-0
2	0-10-16-4-0
3	0-17-2-11-20-0
4	0-7-19-18-8-0
5	0-14-1-13-0
6	0-6-5-0
7	0-12-3-0

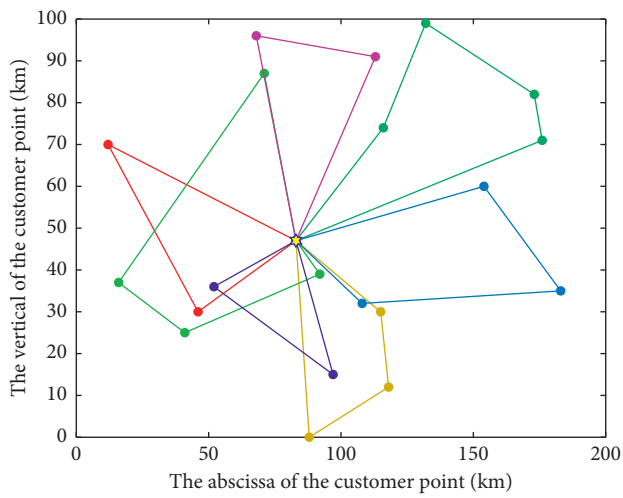


Figure 8: The path graph corresponding to the optimal value of ACSO.

represents the distribution center, and "Point" indicates the coordinates of the location of the customer point.

6. Conclusions

In the study, based on the benefits of CSO and APSO, an Adaptive Cat Swarm Optimization (ACSO) algorithm is proposed. Through the cat swarm behaviour in the tracing mode, there is an adaptive adjustment to its parameters. The effectiveness of it has been tested through 23 benchmark functions. This experimental result indicates that ACSO has excellent performance than other existing heuristics in the process of exploration and exploitation.

In the end, ACSO is applied to VRP. Numerical assessments on four algorithms (ACSO, CSO, APSO, and PSO) reveal that the best result comes from the proposed ACSO algorithm, which further confirms the practicability and effectiveness of the algorithm. However, during the course of the algorithm, because the evolutionary state needs to be evaluated based on the distance to adjust the adaptive parameters in the tracing mode, it requires much more processing time to make related parameter adjustments. Therefore, the future work is to reduce the running time of the algorithm more reasonably without affecting the group to find the optimal solution.

Data Availability

All data are included within the tables of this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61872085), the Natural Science Foundation of Fujian Province (2018J01638), and the Fujian Provincial Department of Science and Technology (2018Y3001).

References

[1] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2008.

[2] H. Wang, S. Rahnamayan, H. Sun, and M. G. Omran, "Gaussian bare-bones differential evolution," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 634–647, 2013.

[3] Z. Meng, J.-S. Pan, and K.-K. Tseng, "PaDE: an enhanced differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80–99, 2019.

[4] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 287–297, 1999.

[5] J. S. Pan, F. R. McInnes, and M. A. Jack, "Application of parallel genetic algorithm and property of multiple global optima to VQ codevector index assignment for noisy channels," *Electronics Letters*, vol. 32, no. 4, pp. 296-297, 1996.

[6] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks 1995*, pp. 1942–1948, Perth, Australia, December 1995.

[7] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, vol. 3, IEEE, Washington, DC, USA, pp. 1945–1950, July 1999.

[8] Y. Shi and Eberhart, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, IEEE, Seoul, Korea, pp. 81–86, May 2001.

[9] S.-C. Chu, P.-w. Tsai, and J.-S. Pan, "Cat swarm optimization," in *Proceedings of the 2006 Pacific Rim International Conference on Artificial Intelligence*, pp. 854–858, Springer, Guilin, China, August 2006.

[10] P. W. Tsai, J. S. Pan, S. M. Chen, B. Y. Liao, and S. P. Hao, "Parallel cat swarm optimization," in *Proceedings of the 2008 International Conference on Machine Learning and Cybernetics*, vol. 6, IEEE, Kunming, China, pp. 3328–3333, 2008.

[11] P.-W. Tsai, J.-S. Pan, S.-M. Chen, and B.-Y. Liao, "Enhanced parallel cat swarm optimization based on the Taguchi method," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6309–6319, 2012.

[12] M. Dorigo and M. Birattari, *Ant Colony Optimization*, Springer, Berlin, Germany, 2010.

[13] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances," in *Handbook of Metaheuristics*, pp. 311–351, Springer, Berlin, Germany, 2019.

[14] S.-C. Chu, J. F. Roddick, C.-J. Su, and J.-S. Pan, "Constrained ant colony optimization for data clustering," in *Proceedings of the 2004 Pacific Rim International Conference on Artificial*

Intelligence, pp. 534–543, Springer, Auckland, New Zealand, August 2004.

[15] S.-C. Chu, J. F. Roddick, and J.-S. Pan, "Ant colony system with communication strategies," *Information Sciences*, vol. 167, no. 1–4, pp. 63–76, 2004.

[16] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization*, pp. 65–74, Springer, Berlin, Germany, 2010.

[17] T.-K. Dao, T.-S. Pan, T.-T. Nguyen, and J.-S. Pan, "Parallel bat algorithm for optimizing makespan in job shop scheduling problems," *Journal of Intelligent Manufacturing*, vol. 29, no. 2, pp. 451–462, 2018.

[18] J.-S. Pan, P. Hu, and S.-C. Chu, "Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power," *Processes*, vol. 7, no. 11, p. 845, 2019.

[19] P. Hu, J.-S. Pan, S.-C. Chu, Q.-W. Chai, T. Liu, and Z.-C. Li, "New hybrid algorithms for prediction of daily load of power network," *Applied Sciences*, vol. 9, no. 21, p. 4514, 2019.

[20] P. Toth and D. Vigo, *The Vehicle Routing Problem*, SIAM, Philadelphia, PA, USA, 2002.

[21] C.-H. Chen, F.-J. Hwang, and H.-Y. Kung, "Travel time prediction system based on data clustering for waste collection vehicles," *IEICE Transactions on Information and Systems*, vol. E102.D, no. 7, pp. 1374–1383, 2019.

[22] C.-H. Chen, "An arrival time prediction method for bus system," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4231-4232, 2018.

[23] C.-H. Chen, "A cell probe-based method for vehicle speed estimation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E103.A, no. 1, pp. 265–267, 2020.

[24] J. S. Pan, L. Kong, T. W. Sung, P. W. Tsai, and V. Snášel, "A clustering scheme for wireless sensor networks based on genetic algorithm and dominating set," *Journal of Internet Technology*, vol. 19, pp. 1111–1118, 2018.

[25] J. S. Pan, L. Kong, T. W. Sung, P. W. Tsai, and V. Snášel, "α-fraction first strategy for hierarchical model in wireless sensor networks," *Journal of Internet Technology*, vol. 19, pp. 1717–1726, 2018.

[26] T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network," *IEEE Access*, vol. 7, pp. 75985–75998, 2019.

[27] C.-I. Wu, H.-Y. Kung, C.-H. Chen, and L.-C. Kuo, "An intelligent slope disaster prediction and monitoring system based on WSN and ANP," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4554–4562, 2014.

[28] J. Wang, X. Gu, W. Liu, A. K. Sangaiah, and H. J. Kim, "An empower Hamilton loop based data collection algorithm with mobile agent for WSNs," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1–14, 2019.

[29] J. Wang, C. Ju, Y. Gao, A. K. Sangaiah, and G. j. Kim, "A PSO based energy efficient coverage control algorithm for wireless sensor networks," *Computers, Materials and Continua*, vol. 56, pp. 433–446, 2018.

[30] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H. J. Kim, "An intelligent data gathering schema with data fusion supported for mobile sink in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 15, no. 3, 2019.

[31] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.

[32] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1362–1381, 2009.

[33] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, IEEE, Piscataway, NJ, USA, pp. 39–43, 1995.

[34] M. Molga and C. Smutnicki, "Test functions for optimization needs," *Test Functions for Optimization Needs*, vol. 101, 2005.

[35] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[36] M. Gendreau, A. Hertz, and G. Laporte, "A tabu search heuristic for the vehicle routing problem," *Management Science*, vol. 40, no. 10, pp. 1276–1290, 1994.

[37] P. Toth and D. Vigo, "The granular tabu search and its application to the vehicle-routing problem," *INFORMS Journal on Computing*, vol. 15, no. 4, pp. 333–346, 2003.

[38] A. Van Breedam, "Improvement heuristics for the vehicle routing problem based on simulated annealing," *European Journal of Operational Research*, vol. 86, no. 3, pp. 480–490, 1995.

[39] W.-C. Chiang and R. A. Russell, "Simulated annealing metaheuristics for the vehicle routing problem with time windows," *Annals of Operations Research*, vol. 63, no. 1, pp. 3–27, 1996.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 8184254, 10 pages
https://doi.org/10.1155/2020/8184254

Research Article

Sine Cosine Algorithm with Multigroup and Multistrategy for Solving CVRP

Qingyong Yang, Shu-Chuan Chu, Jeng-Shyang Pan, and Chien-Ming Chen

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong 266590, China

Correspondence should be addressed to Jeng-Shyang Pan; jengshyangpan@gmail.com

Received 3 February 2020; Accepted 15 February 2020; Published 27 March 2020

Guest Editor: Chi-Hua Chen

Sine Cosine Algorithm (SCA) has been proved to be superior to some existing traditional optimization algorithms owing to its unique optimization principle. However, there are still disadvantages such as low solution accuracy and poor global search ability. Aiming at the shortcomings of the sine cosine algorithm, a multigroup multistrategy SCA algorithm (MMSCA) is proposed in this paper. The algorithm executes multiple populations in parallel, and each population executes a different optimization strategy. Information is exchanged among populations through intergenerational communication. Using 19 different types of test functions, the optimization performance of the algorithm is tested. Numerical experimental results show that the performance of the MMSCA algorithm is better than that of the original SCA algorithm, and it also has some advantages over other intelligent algorithms. At last, it is applied to solving the capacitated vehicle routing problem (CVRP) in transportation. The algorithm can get better results, and the practicability and feasibility of the algorithm are also proved.

1. Introduction

With the rapid development of human society and the continuous improvement of computer technology, people are facing more and more optimization problems in scientific research and engineering practice. The optimization problems can be understood as choosing a most suitable solution among many options for solving a complex problem without changing the constraints. However, with the increasing scale of the problem, the computing time also shows an exponential growth trend. When solving large-scale optimization problems, traditional algorithms seem powerless. For quite some time, people have been working hard to find new ways to better solve optimization problems. With the robustness, parallelism, and intelligence of its methods, computational intelligence (CI) can well solve many optimization problems, which has attracted the attention of many scholars. Also, computational intelligence (CI) plays an important role in scientific research and production practice.

Computational intelligence is a general term for a class of algorithms that are inspired by natural and human wisdom. Computational intelligence includes three basic areas: fuzzy computing [1–3], neural networks [4, 5], and evolutionary computing [6]. Evolutionary computing is one of the important research fields of intelligent computing. Up to now, scholars have put forward many excellent intelligent optimization algorithms. Typical examples are as follows: genetic algorithm (GA) by simulating the genetic evolution of the biological world [7]; particle swarm optimization (PSO) by simulating bird swarm predation [8–10]; and ant colony algorithm (ACO) by simulating ant colony foraging behavior [11, 12]. There are many intelligent optimization algorithms proposed by simulating other phenomena in nature, such as cat swarm algorithm (CSO) [13–15], artificial bee colony algorithm (ABC) [16, 17], differential evolution algorithm (DE) [18–21], multiverse optimizer (MVO) [22, 23], flower pollination algorithm [24, 25], gray wolf algorithm (GWO) [26–28], pigeon-inspired algorithm (PIO)

[29, 30], bat algorithm (BA) [31, 32], symbiotic organism search algorithm (SOS) [33, 34], and QUATRE [35–37].

At present, there are so many optimization algorithms, which can solve some complex optimization problems well. Why do we need so many optimization algorithms? As demonstrated by the No Free Lunch (NFL) [38] proposed by Wolpert and Macready, no single optimization algorithm is applicable to all problems. Inspired by this, a new intelligent optimization algorithm was proposed by Australian scholar Mirjalili in 2016, which is called Sine Cosine Algorithm (SCA) [39]. The SCA algorithm iterates through the properties of the sine and cosine functions to achieve optimization. It has fewer parameter settings, is easy to implement, and has a strong optimization ability. It has been proved that it is better than PSO algorithm, genetic algorithm (GA), and firefly algorithm (FA) in convergence with accuracy and speed [39].

With the rapid development of software and hardware, parallel computing has become a form of high-performance computing. In evolutionary computing, parallelism often represents the iterative updating of multiple populations at the same time. The advantage of this method is to ensure population diversity, to further improve the search ability and performance of the algorithm. Especially when solving complex optimization problems, parallelizing the algorithm is an effective way to improve the efficiency and accuracy of the algorithm. At present, many existing algorithms have successfully applied the parallel mechanism, such as parallel PSO [40], parallel ACO [41], and parallel QUATRE [42]. Inspired by this, this paper introduces the multigroup and multistrategy optimization mechanism to further improve the SCA algorithm. It is called MMSCA. When the algorithm is solved, multiple populations execute in parallel, and each population adopts different updating strategies. By comparing the results of test functions, MMSCA is better than SCA in solving accuracy and convergence ability and has certain advantages over other intelligent optimization algorithms.

At present, transportation is gradually moving towards intelligent transportation. One of the important components of an intelligent transportation system is wireless sensor network technology (WSN). With the help of wireless sensor network technology [43–48], the safety, unblocking, and environmental protection problems that plague modern transportation can be fundamentally alleviated. However, there are still many transportation problems to be solved. The capacitated vehicle routing problem (CVRP) in transportation problems is taken as the application object of MMSCA in this paper. On the one hand, the practicability of the algorithm is further confirmed. On the other hand, it brings a new solution to CVRP problem.

The remaining chapters of the article are arranged as follows: the original SCA algorithm and related improvements around it are reviewed in Section 2. Section 3 describes the MMSCA design process in detail. The results of numerical experiments are presented and discussed in Section 4. How to apply MMSCA to CVRP is covered in Section 5. At last, Section 6 is a summary of the article and an outlook for future work.

2. Related Works

2.1. Original Sine Cosine Algorithm. In the SCA algorithm, a population X is initialized first, where the population X contains m individuals, and each individual has n dimensions. The fitness function is used to evaluate each individual, and the best individual position $P_b = (P_{b1}, P_{b2}, \ldots, P_{bn})^T$ among all the current individuals is selected. Then, the algorithm iterates according to the set number of iterations. During the iteration, the position of the individual is updated. The update formula is as follows:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times \left| r_3 \times P_b^t - X_i^t \right|, & r_4 < 0.5, \\ X_i^t + r_1 \times \cos(r_2) \times \left| r_3 \times P_b^t - X_i^t \right|, & r_4 \geq 0.5, \end{cases} \tag{1}$$

where X_i^t represents the position of the ith ($i = 1, 2, \ldots, m$) individual in the tth iteration; $r_2 \in [0, 2\pi]$, $r_3 \in [0, 2]$, and $r_4 \in [0, 1]$ are three random numbers. P_b^t represents the best individual position after t iterations. r_1 is the control parameter, which balances the exploration and exploitation stages of the algorithm. The updated formula is as follows:

$$r_1 = a - a\frac{t}{T}, \tag{2}$$

where a is a constant, t is the number of current iterations, and T is the maximum number of iterations.

Since equation (1) uses sine and cosine functions, the algorithm is named Sine Cosine Algorithm (SCA). Four important parameters play different roles in the process of iterative optimization of the algorithm: the parameter r_1 is used as a control parameter to determine the moving direction of the individual position in the next iteration; the parameter r_2 determines how far to move in the next iteration; as a random weight, parameter r_3 affects the role of the optimal individual in the next iteration; the parameter r_4 enables the algorithm to iteratively update the individual positions in the sine and cosine parts with equal probability. Like other intelligent optimization algorithms, there need to be two phases: local exploitation and global exploration. In SCA algorithm, when $r_1 \sin(r_2)$ or $r_1 \cos(r_2)$ function value is between −1 and 1, the algorithm performs local exploitation; when $r_1 \sin(r_2)$ or $r_1 \cos(r_2)$ function value is less than −1 or greater than 1, the algorithm performs global exploration.

2.2. Related Improvements to the SCA Algorithm. With its unique optimization principles and theoretical advantages, the SCA algorithm has been proved to be superior to PSO, genetic algorithm (GA), and other existing algorithms in the most of the examples and has been applied in many practical problems. However, there are still some disadvantages such as low solution accuracy and poor global search ability. This is because when the SCA algorithm updates the individual location, the next generation location of the individual is only expanded around the current optimal individual location. In particular, with the decrease of the diversity of the population in the later period of the algorithm, the global

search ability becomes worse, which easily leads to the algorithm falling into the local optimal trap.

In view of the shortcomings of the SCA algorithm, scholars at home and abroad have made some improvements. Some scholars introduced search strategies of other algorithms or improved the setting of the parameters of the SCA algorithm itself: Long et al. [49] introduced nonlinear weight factor and inertial weight based on Gaussian distribution, respectively, to improve the ability to avoid falling into the trap of local optimization and convergence speed; Qu et al. [50] proposed an improved SCA algorithm based on neighborhood search and Greedy Levy Mutation to better balance the phases of local exploitation and global exploration of the algorithm. Some scholars have also integrated the SCA algorithm with other algorithms to further improve the optimization ability of the algorithm: Chegini et al. [51] proposed to mix the SCA algorithm with the PSO algorithm to improve the global search capability of the algorithm and the calculation accuracy; Nenavath and Jatoth [52] mixed the SCA algorithm and the DE algorithm to further improve the convergence speed of the algorithm and the ability to avoid local optimization.

There are many improvements to the SCA algorithm, and the improvement methods can be summarized into the abovementioned two categories. Aiming at the shortcomings of the SCA algorithm, this paper adopts a multigroup and multistrategy optimization mechanism to further improve the accuracy of the solution and the global search ability of the algorithm. The next section will detail how to improve the SCA algorithm.

3. Sine Cosine Algorithm with Multigroup and Multistrategy (MMSCA)

This section describes in detail how to apply the multigroup and multistrategy mechanism to improve the Sine Cosine Algorithm (SCA).

3.1. Algorithm Idea. The use of a single population in basic SCA algorithm inevitably leads to the problem of falling into the local optimal trap. At the same time, when the algorithm updates individual location in each iteration, the location of candidate individuals is only expanded around the global optimal location, which results in the strong local search capability of the basic SCA algorithm and the weak global search capability. Based on the shortcomings of the abovementioned basic SCA algorithm, the concept of multigroup and multistrategy is introduced to enhance the optimization ability of SCA algorithm. In the multigroup multistrategy SCA algorithm (MMSCA), each population has the same number of individuals and uses a different update strategy. Before a certain number of generations, each population is optimized according to its own strategy, and the population does not interfere with each other. After reaching a certain number of generations, the populations communicate with each other and perform the optimal individual migration between the populations. The worst individual in each population will be replaced by the optimal individual in the other populations. Repeat the abovementioned steps until the algorithm iterates to the maximum number of iterations or the optimal solution is found.

In this article, the update strategies are divided into two types: rand strategy and best strategy. In the rand strategy, the roulette wheel selection scheme in the genetic algorithm (GA) is borrowed, and the optimal individual in the group is no longer selected as the solution target of other individuals. Instead, the algorithm uses multiple roulette methods to choose the next generation of solving goals to increase the diversity of population solutions. In this way, the ability of global optimization can be enhanced and the possibility of local optimal trap will be increased. In the best strategy, the best individual in all populations is still the solution target. The purpose of doing this is to maintain the strong local optimization ability of the original algorithm and avoid the instability of the algorithm caused by the introduction of the rand strategy.

The update formula is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} + r_1 \times \sin(r_2) \times \left| r_3 \times P_b^t - X_{i,j}^t \right|, & r_4 < 0.5, \\ X_{i,j}^{t} + r_1 \times \cos(r_2) \times \left| r_3 \times P_b^t - X_{i,j}^t \right|, & r_4 \geq 0.5, \end{cases} \tag{3}$$

where $X_{i,j}^t$ represents the position of the jth individual in the ith population in the tth iteration and P_b^t represents the selected target individual after multiple generations of roulette wheel selection in the rand strategy executing population. Also, in the population that executes the best strategy, it represents the globally optimal individual.

3.2. Algorithm Steps. Based on the abovementioned description, the specific steps of the MMSCA algorithm are given below:

Step 1: randomly generate N (N is an even number) populations of size m, each population having n dimensions. Initialize the maximum number of iterations, independent evolution algebra, iteration counter etc.

Step 2: evaluate each population and calculate the fitness value of each individual in each population. The population with odd group numbers executes the rand strategy, and the target individual is selected using the roulette wheel selection scheme; the population with even group numbers executes the best strategy, sorts and selects the best individual.

Step 3: use formula (3) to update the individual positions in the population.

Step 4: if the communication conditions are met, interpopulation communication is conducted. The worst individual in a population is replaced by the best individual in another population. If the communication condition is not met, return to step 2.

Step 5: if the algorithm iterates to the maximum number of iterations or the optimal individual is found

to meet the convergence conditions, the algorithm is terminated; otherwise return to step 2.

The algorithm flow chart is shown in Figure 1.

4. Results and Discussion

4.1. Comparison with SCA Algorithm. In order to test the influence of multigroup and multistrategy mechanism on SCA algorithm, 19 standard test functions ($f_1 - f_7$ as unimodal benchmark function, $f_8 - f_{13}$ as multimodal benchmark function, and $f_{14} - f_{19}$ as composite benchmark function) in the SCA algorithm literature [39] are used for testing. The minimum value of test function $f_1 - f_{15}$ is 0 except f_8, f_8 and $f_{16} - f_{19}$ are −2094.9145, −1.036.0.398, 3, −3.86, respectively. For the fairness of comparison, the maximum number of iterations is 1000 and each population size is set to 30. In the proposed algorithm, the independent evolutionary algebra is set to 50 and the number of groups is two. For each test function, each algorithm is run independently for 30 times and the average value is calculated. Table 1 records the experimental results, with the best optimization results in bold. All simulation experiments are implemented on the same computer, and the programming tool is Matlab2018a.

As shown in the abovementioned table, in the test functions $f_1 - f_{15}$, except f_8, the MMSCA algorithm has better solution accuracy than the original SCA algorithm. For function $f_{16} - f_{19}$, both algorithms show the same optimization performance, and the results are close to or reach the expected minimum. In general, the optimization ability of the SCA algorithm has been improved to some extent after the introduction of the multigroup and multistrategy mechanism. Figure 2 shows the convergence curves of the two algorithms on the unimodal functions f_2 and f_3, the multimodal functions f_9 and f_{11}, and the composite function f_{14}. As can be seen from the figure, MMSCA algorithm has also significantly improved the convergence speed compared with SCA algorithm.

4.2. Comparison with Other Intelligent Optimization Algorithms. To further verify the effectiveness of the MMSCA algorithm, this article also compares it with other three intelligent optimization algorithms: the PSO algorithm, the parallel PSO algorithm, and the DE algorithm. The population size is set to 30. 1000 is the maximum number of iterations. For each test function, each algorithm is tested 30 times and the average value is obtained. The other parameters of other algorithms are shown in Table 2. The test results of MMSCA and other three intelligent optimization algorithms on 19 test functions are shown in Table 3. The bold type in the table indicates the best results of all comparison algorithms.

As shown in Table 3, compared with the PSO algorithm, PPSO algorithm, and DE algorithm, the MMSCA algorithm has obtained better results on 11 test functions. The MMSCA algorithm has the same test results on the test functions f_6 and $f_{18} - f_{19}$ as the other three algorithms, and they all obtain the expected minimum. However, the test results on the test functions f_6 and f_{13} are significantly worse than those of the other three optimization algorithms; the MMSCA algorithm obtains better results on the test function f_{11}, and obtains a minimum value of 0.

Based on the comparison results of the two subsections, compared with the original SCA algorithm, the optimization performance of the algorithm is improved after the introduction of multigroup and multistrategy mechanism;

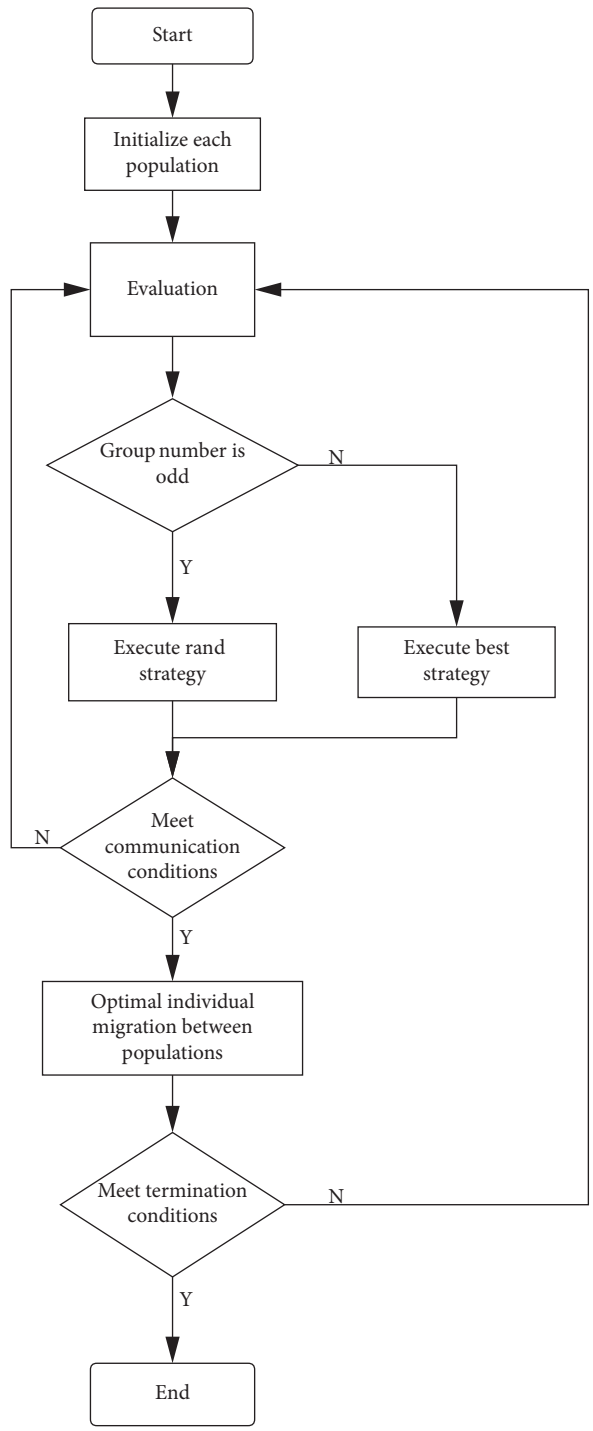


Figure 1: Flowchart of sine cosine algorithm with multigroup and multistrategy.

Table 1: Comparison of test results with SCA algorithm.

Function	SCA	MMSCA
f_1	$5.86E-28$	**2.95E − 30**
f_2	$2.19E-17$	**8.02E − 20**
f_3	$1.46E-08$	**6.73E − 30**
f_4	$1.31E-07$	**1.56E − 08**
f_5	$7.05E+00$	**7.02E + 00**
f_6	$3.73E-01$	**2.98E − 01**
f_7	$1.82E-03$	**1.83E − 04**
f_8	**−2.22E + 03**	$-2.29E+03$
f_9	$6.83E-01$	**3.30E − 06**
f_{10}	$6.48E-14$	**2.91E − 14**
f_{11}	$2.34E-02$	**0.00E + 00**
f_{12}	$6.92E-02$	**5.60E − 02**
f_{13}	$2.68E-01$	**2.59E − 01**
f_{14}	$1.86E+00$	**9.98E − 01**
f_{15}	$9.20E-04$	**6.85E − 04**
f_{16}	**−1.03E + 00**	**−1.03E + 00**
f_{17}	**3.99*E* − 01**	**3.99*E* − 01**
f_{18}	**3.00E + 00**	**3.00E + 00**
f_{19}	**−3.86E + 00**	**−3.86E + 00**

compared with other intelligent optimization algorithms, it also shows strong competitiveness. So as to further study the feasibility and practicability of the MMSCA algorithm in processing practical problems, this paper selects the capacitated vehicle routing problem (CVRP) in transportation as the application object. The next section will describe in detail how to apply the MMSCA algorithm and solve it.

5. Applying MMSCA to Solve CVRP

VRP (Vehicle Routing Problem) has always been the focus of research in the field of transportation. At the same time, there are many applications in real life, including express delivery, air transportation, bus route planning, and so on. VRP was first proposed by scholars Dantzing and Ramser [53] in the late 1950s. Since it was proposed, it has received extensive attention from scholars at home and abroad, and scholars have also carried out a lot of fruitful work on it. After years of research and development, more models of VRP have been derived, and the solving algorithms have become more diverse. Solution algorithms can be divided into two categories: the first type is accurate algorithm and the second is approximate algorithm or heuristic algorithm. Since the VRP problem has been proved to be an NP-Hard problem, the second kind of solution algorithm is the main method to solve it [54]. The GA algorithm, the ACO algorithm, the SA (simulated annealing) algorithm, the PSO algorithm, and other algorithms have been successfully applied to solve VRP problems and achieved good results. In this paper, CVRP (Capacitated Vehicle Routing Problem) is selected as the application object of MMSCA algorithm to further verify the practicability of the algorithm.

5.1. CVRP Problem Description and Model. Problem description: a central warehouse has k distribution vehicles, and the maximum load capacity of each distribution vehicle is Q. Now, we need to transport and distribute materials to n customers (nodes), how to arrange a certain number of vehicles to meet the needs of these customers, and make the total driving distance of vehicles shortest. The following constraints must also be met:

(1) All delivery vehicles start from the central warehouse and eventually return to the central warehouse

(2) Each delivery vehicle is not allowed to exceed the maximum load capacity Q

(3) Each customer can only be served once, and each vehicle can only take one route

For the convenience of description, the following symbols are defined:

D: total length of all vehicle paths

k: total number of vehicles

n: total number of customers

v: vehicle number (1, 2, ..., k)

$$x_{ijv} = \begin{cases} 1, & \text{if vehicle } v \text{ travels from customer } i \text{ to customer } j, \\ 0, & \text{else,} \end{cases}$$
$$y_{iv} = \begin{cases} 1, & \text{if customer } i \text{ needs to be fulfilled by vehicle } v, \\ 0, & \text{else,} \end{cases} \tag{4}$$

c_{ij}: the distance between customer point i and j

g_i: the demand of the i-th customer ($i = 1, 2, \ldots, n$)

Q: the maximum loading capacity of the vehicle

The central warehouse number is 0, and the customer number is i ($i = 1, 2, \ldots, n$). The CVRP mathematical model is expressed as follows:

Objective function:

$$\min D = \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{v=0}^{k} c_{ij}x_{ijv}. \tag{5}$$

Constraint conditions:

$$\sum_{i=0}^{n} x_{ijv} = y_{jv}, \quad j = 1, 2, \ldots, n;\ v = 1, 2, \ldots, k, \tag{6}$$

$$\sum_{j=0}^{n} x_{ijv} = y_{jv}, \quad i = 1, 2, \ldots, n;\ v = 1, 2, \ldots, k, \tag{7}$$

$$\sum_{i=0}^{n} g_i y_{iv} \le Q, \quad v = 1, 2, \ldots, k, \tag{8}$$

$$\sum_{v=1}^{k} y_{iv} = 1, \quad i = 1, 2, \ldots, n. \tag{9}$$

Among them, equations (6) and (7) are traffic flow constraints, requiring a car to leave a customer node after it

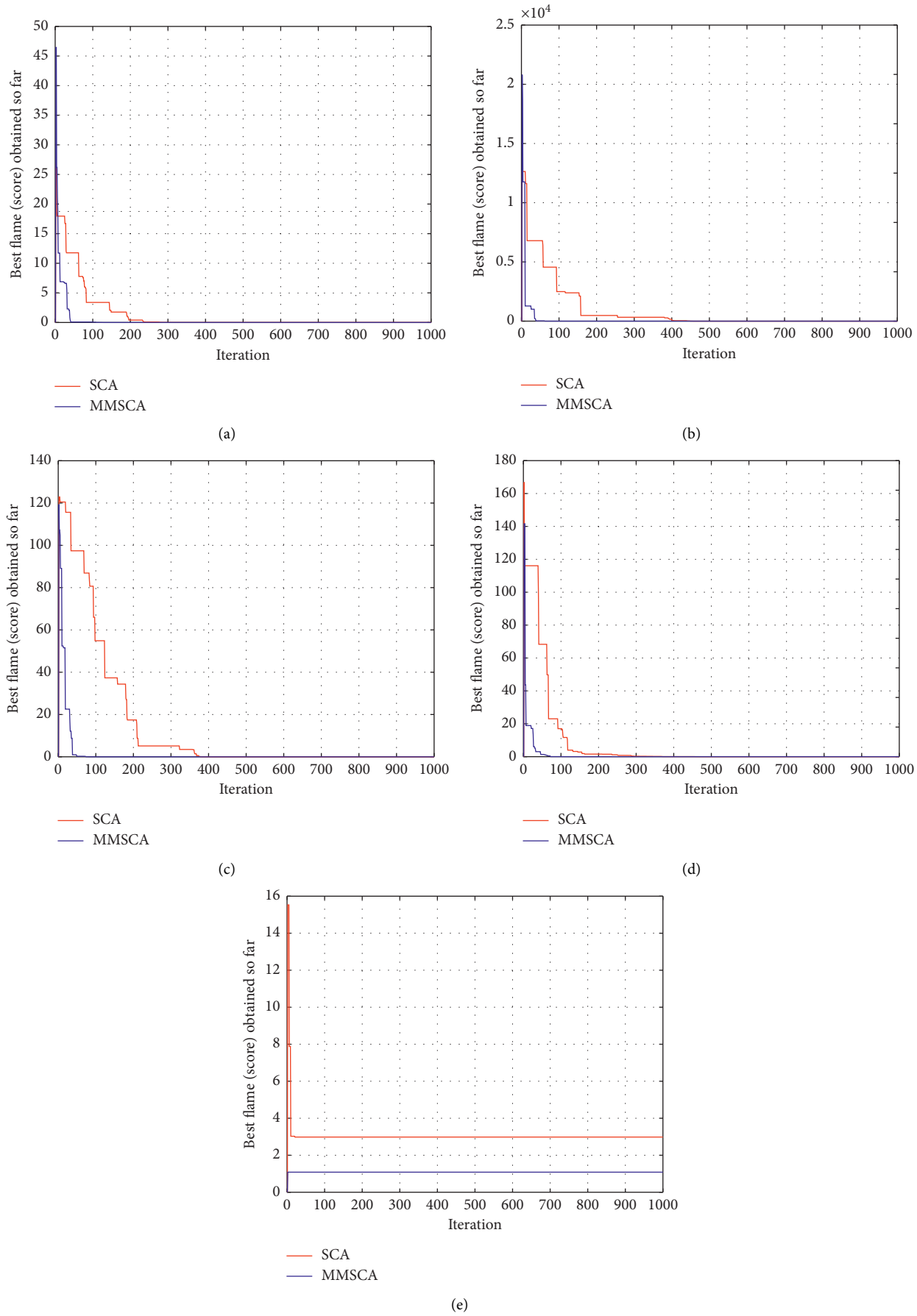


Figure 2: Convergence curve of benchmark functions: f_2, f_3, f_9, f_{11}, f_{14}. (a) f_2. (b) f_3. (c) f_9. (d) f_{11}. (e) f_{14}.

Table 2: Parameter settings.

Algorithm	Parameters
PSO	$c1 = c2 = 2,\ \omega = 0.9$
PPSO	The number of groups is 2, $c1 = c2 = 2,\ \omega = 0.7$
DE	$\mathrm{pCR} = 0.2,\ \beta_{\max} = 0.8,\ \beta_{\min} = 0.2$

Table 3: Test results compared with other optimization algorithms.

Function	MMSCA	PSO	PPSO	DE
f_1	**2.95E − 30**	$6.92E-03$	$3.03E-28$	$8.14E-15$
f_2	**8.02E − 20**	$2.46E-01$	$5.93E-13$	$1.58E-08$
f_3	**6.73E − 30**	$8.03E-02$	$1.08E-25$	$6.38E+01$
f_4	**1.56E − 08**	$7.92E-02$	$6.67E-08$	$9.47E-02$
f_5	$7.02E+00$	$1.70E+01$	**1.16E + 00**	$2.94E+01$
f_6	$2.98E-01$	$6.48E-03$	$1.50E-09$	**6.89E − 15**
f_7	**1.83E − 04**	$1.30E-02$	$2.07E-03$	$2.87E-02$
f_8	**−2.29E + 03**	$-2.40E+03$	$-2.70E+03$	$-1.18E+02$
f_9	**3.30E − 06**	$3.34E+00$	$6.70E+00$	$5.81E+01$
f_{10}	**2.91E − 14**	$2.77E-01$	$3.68E-13$	$5.08E-08$
f_{11}	**0.00E + 00**	$2.29E-01$	$3.32E-01$	$9.88E-15$
f_{12}	$5.60E-02$	$6.44E-02$	$1.04E-02$	**2.25E − 16**
f_{13}	$2.59E-01$	$7.68E-03$	$2.28E-03$	**1.86E − 15**
f_{14}	**9.98E − 01**	$3.86E+00$	$1.99E+00$	$3.26E+00$
f_{15}	**6.85E − 04**	$4.52E-03$	$1.80E-03$	$7.23E-04$
f_{16}	**−1.03E + 00**	**−1.03E + 00**	**−1.03E + 00**	**−1.03E + 00**
f_{17}	$3.99E-01$	$3.99E-01$	$3.99E-01$	**3.98E − 01**
f_{18}	**3.00E + 00**	**3.00E + 00**	**3.00E + 00**	**3.00E + 00**
f_{19}	**−3.86E + 00**	**−3.86E + 00**	**−3.86E + 00**	**−3.86E + 00**

Table 4: Comparison of MMSCA and other two algorithms.

Data set	Best known results	MMSCA		PSO		GA	
		Best	Dev (%)	Best	Dev (%)	Best	Dev (%)
A-n32-k5	784	808	3.06	829	5.73	818	4.34
A-n33-k5	661	690	4.39	705	6.65	674	1.97
A-n39-k6	831	868	4.45	872	6.08	866	5.35
A-n46-k7	914	964	5.47	977	6.89	957	4.7
A-n60-k9	1408	1426	1.28	1476	4.83	1410	0.14

has completed service; formula (8) represents the vehicle capacity constraint and indicates that the total customer demand on the route served by vehicle v cannot exceed the vehicle loading capacity Q; formula (9) ensures that each customer can only be served by one vehicle.

5.2. MMSCA Algorithm for CVRP. The MMSCA algorithm is a continuous space algorithm, but the CVRP problem is an integer programming problem, so it needs to be modified in the implementation process.

(1) Encoding and decoding

Encoding: using natural number coding, each individual in the population corresponds to a unique arrangement from 1 to n, and each individual corresponds to a distribution scheme.

Decoding: arrange the vehicles according to the sequence of customers in the distribution scheme and the constraint of maximum loading capacity. That is, it is sequentially included in the path of the first car according to the customer's coding order, and the second car's path is arranged until the maximum load of the first car is exceeded.

(2) Evaluation criteria

The total length of the vehicle path is used to evaluate all individuals and the merits of each distribution scheme.

(3) Implementation process

According to the abovementioned coding method, each population is generated. Each individual (distribution scheme) is evaluated according to the evaluation criteria. Update each individual with formula (3) to generate a new distribution scheme. According to the constraint conditions, the actual vehicle path is generated and evaluated. In the process of updating with formula (3), integer

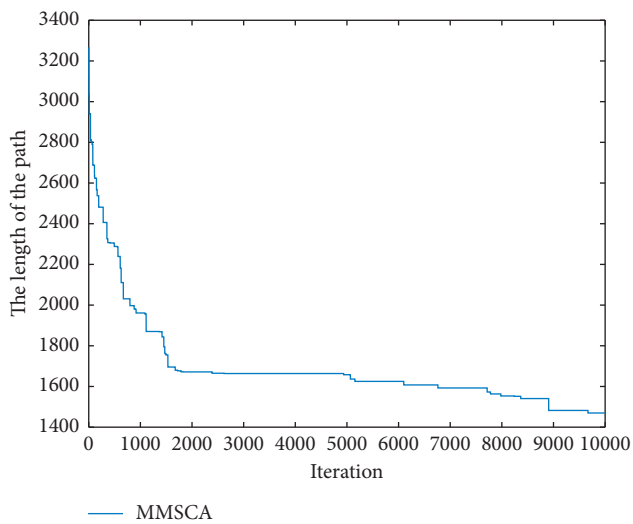


Figure 3: Convergence curve (A-n60-k9).

specification constraints are required to avoid generating infeasible solutions.

5.3. Simulation and Results Analysis. In this paper, five standard VRPLIB test samples are selected for testing, and the test results of the MMSCA algorithm are compared with the PSO algorithm [55] and the GA algorithm. Table 4 shows the experimental results. In the experiment, the population number of MMSCA algorithm is divided two groups, the number of individuals in each group is 50, the solution is thirty times, and the number of iterations is 10,000. During the solution process, the best solution result is recorded.

Where the Best column represents the known optimal results, the Dev column indicates the deviation between the optimal solutions obtained by the algorithm and the best known results. As can be seen from the abovementioned table, the MMSCA algorithm has achieved better results and has certain advantages compared with the PSO algorithm and the genetic algorithm. But, there is still a certain gap with the currently known optimal solutions, indicating that the algorithm still has room for improvement. Figure 3 shows the convergence curve when the MMSCA algorithm solves the CVRP instance A-n60-k9.

6. Conclusion

Firstly, in view of the problems existing in the SCA algorithm update process, this paper proposes multigroup multistrategy SCA algorithm (MMSCA). The algorithm contains multiple populations with the same number of individuals, and each population performs a different update strategy. Multiple strategies include rand strategy and best strategy. The rand strategy can improve the global optimization ability of the algorithm, and the algorithm can still be disturbed at the later stage to avoid algorithm stagnation. The best strategy is responsible for keeping the algorithm stable and speeds up the convergence speed of the algorithm. Through numerical experiments on the test function, the MMSCA algorithm has a certain improvement in the accuracy and convergence speed of the original SCA algorithm. Finally, this paper successfully applies the MMSCA algorithm to solve CVRP (Capacitated Vehicle Routing Problem) and obtains better results, which further confirms the practicability and effectiveness of the algorithm. The next research direction will continue to improve the algorithm and apply it to more practical engineering problems.

Data Availability

All data are included within the tables of this article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] F. Ahmed and K. Kilic, "Fuzzy Analytic Hierarchy Process: a performance analysis of various algorithms," *Fuzzy Sets and System*, vol. 362, pp. 110–128, 2019.

[2] S.-M. Chen, G. M. T. Manalu, J.-S. Pan, and H.-C. Liu, "Fuzzy forecasting based on two-factors second-order fuzzy-trend logical relationship groups and particle swarm optimization techniques," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1102–1117, 2013.

[3] S.-M. Chen, Y.-C. Chang, and J.-S. Pan, "Fuzzy rules interpolation for sparse fuzzy rule-based systems based on interval type-2 Gaussian fuzzy sets and genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 412–425, 2013.

[4] R. R. Sarukkai, "Link prediction and path analysis using Markov chains," *Computer Networks*, vol. 33, no. 1–6, pp. 377–386, 2000.

[5] S. C. Chu, X. Xue, J. S. Pan et al., "Optimizing ontology alignment in vector space," *Journal of Internet Technology*, vol. 21, no. 1, pp. 15–22, 2020.

[6] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2016.

[7] J. H. Holland and J. S. Reitman, "Cognitive systems based on adaptive algorithms," *ACM SIGART Bulletin*, vol. 63, pp. 49–49, 1977.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95—International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, 1995.

[9] C. Sun, J. Zeng, J. Pan, S. Xue, and Y. Jin, "A new fitness estimation strategy for particle swarm optimization," *Information Sciences*, vol. 221, pp. 355–370, 2013.

[10] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Information Sciences*, vol. 181, no. 20, pp. 4699–4714, 2011.

[11] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[12] S. C. Chu, J. F. Roddick, and J. S. Pan, "Ant colony system with communication strategies," *Information Sciences*, vol. 167, no. 1–4, pp. 63–76, 2004.

[13] S. C. Chu, P. W. Tsai, and J. S. Pan, "Cat swarm optimization," in *Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence*, pp. 854–858, Guilin, China, August 2006.

[14] P. W. Tsai, J. S. Pan, S. M. Chen et al., "Parallel cat swarm optimization," in *Proceedings of the 2008 International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3328–3333, Kunming, China, 2008.

[15] P.-W. Tsai, J.-S. Pan, S.-M. Chen, and B.-Y. Liao, "Enhanced parallel cat swarm optimization based on the Taguchi method," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6309–6319, 2012.

[16] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[17] P. W. Tsai, J. S. Pan, B. Y. Liao et al., "Enhanced artificial bee colony optimization," *International Journal of Innovative Computing, Information & Control: IJICIC*, vol. 5, no. 12, pp. 5081–5092, 2009.

[18] X. M. Hu, J. Zhang, and H. H. Chen, "Optimal vaccine distribution strategy for different age groups of population: a differential evolution algorithm approach," *Mathematical Problems in Engineering*, vol. 2014, Article ID 702973, 7 pages, 2014.

[19] Z. Meng, J.-S. Pan, and K.-K. Tseng, "PaDE: an enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80–99, 2019.

[20] J.-S. Pan, N. Liu, and S.-C. Chu, "A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning," *IEEE Access*, vol. 8, pp. 17691–17712, 2020.

[21] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[22] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2016.

[23] X. P. Wang, J.-S. Pan, and S.-C. Chu, "A parallel multi-verse optimizer for application in multilevel image segmentation," *IEEE Access*, vol. 8, pp. 32018–32030, 2020.

[24] E. Nabil, "A modified flower pollination algorithm for global optimization," *Expert Systems with Applications*, vol. 57, pp. 192–203, 2016.

[25] T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network," *IEEE Access*, vol. 7, pp. 75985–75998, 2019.

[26] E. Emary, W. Yamany, A. E. Hassanien, and V. Snasel, "Multi-objective gray-wolf optimization for attribute reduction," *Procedia Computer Science*, vol. 65, pp. 623–632, 2015.

[27] P. Hu, J.-S. Pan, S.-C. Chu, Q.-W. Chai, T. Liu, and Z.-C. Li, "New hybrid algorithms for prediction of daily load of power network," *Applied Sciences*, vol. 9, no. 21, p. 4514, 2019.

[28] J.-S. Pan, P. Hu, and S.-C. Chu, "Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power," *Processes*, vol. 7, no. 11, p. 845, 2019.

[29] H. Duan and P. Qiao, "Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning," *International Journal of Intelligent Computing and Cybernetics*, vol. 7, no. 1, pp. 24–37, 2014.

[30] A. Q. Tian, S. C. Chu, J. S. Pan et al., "A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station," *Sustainability*, vol. 12, no. 3, 2020.

[31] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284, pp. 65–74, 2010.

[32] P. W. Tsai, J. S. Pan, B. Y. Liao et al., "Bat algorithm inspired algorithm for solving numerical optimization problems," *Applied Mechanics and Materials*, vol. 148, pp. 134–137, 2012.

[33] A. E. Ezugwu and D. Prayogo, "Symbiotic organisms search algorithm: theory, recent advances and applications," *Expert Systems with Applications*, vol. 119, pp. 184–209, 2019.

[34] S.-C. Chu, Z.-G. Du, and J.-S. Pan, "Symbiotic organism search algorithm with multi-group quantum-behavior communication scheme applied in wireless sensor networks," *Applied Sciences*, vol. 10, no. 3, p. 930, 2020.

[35] Z. Meng, J.-S. Pan, and H. Xu, "QUasi-Affine TRansformation Evolution (QUATRE) Algorithm: a cooperative swarm based algorithm for global optimization," *Knowledge-Based Systems*, vol. 109, pp. 104–121, 2016.

[36] Z. Y. Meng and J. S. Pan, "QUasi-Affine TRansformation Evolution (QUATRE) Algorithm: a parameter-reduced differential evolution algorithm for optimization problems," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 4082–4089, Vancouver, Canada, July 2016.

[37] J. S. Pan, Z. Y. Meng, H. R. Xu et al., "QUasi-Affine TRansformation Evolution (QUATRE) Algorithm: a new simple and accurate structure for global optimization," in *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 657–667, Morioka, Japan, August 2016.

[38] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[39] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 809–818, 2016.

[40] J. F. Chang, J. F. Roddick, J. S. Pan et al., "A parallel particle swarm optimization algorithm with communication strategies," *Journal of Information Science and Engineering*, vol. 21, no. 4, pp. 809–818, 2005.

[41] S.-C. Chu, J. F. Roddick, J.-S. Pan, and C.-J. Su, "Parallel ant colony systems," in *Lecture Notes in Computer Science*, vol. 2871, pp. 279–284, Springer, Berlin, Germany, 2003.

[42] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.

[43] C.-I. Wu, H.-Y. Kung, C.-H. Chen, and L.-C. Kuo, "An intelligent slope disaster prediction and monitoring system based on WSN and ANP," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4554–4562, 2014.

[44] H.-W. Wang, C.-H. Chen, D.-Y. Cheng, C.-H. Lin, and C.-C. Lo, "A real-time pothole detection approach for intelligent transportation system," *Mathematical Problems in Engineering*, vol. 2015, Article ID 869627, 7 pages, 2015.

[45] J. S. Pan, L. P. Kong, T. W. Sung et al., "α-fraction first strategy for hierarchical model in wireless sensor networks," *Journal of Internet Technology*, vol. 19, no. 6, pp. 1717–1726, 2018.

[46] J. S. Pan, L. P. Kong, T. W. Sung et al., "A clustering scheme for wireless sensor networks based on genetic algorithm and dominating set," *Journal of Internet Technology*, vol. 19, no. 4, pp. 1111–1118, 2018.

[47] J. Wang, Y. Cao, B. Li, H.-j. Kim, and S. Lee, "Particle swarm optimization based clustering algorithm with mobile sink for WSNs," *Future Generation Computer Systems*, vol. 76, pp. 452–457, 2017.

[48] J. Wang, Y. Gao, X. Yin, F. Li, and H.-J. Kim, "An Enhanced PEGASIS Algorithm with Mobile Sink Support for Wireless Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 9472075, 9 pages, 2018.

[49] W. Long, T. Wu, X. Liang, and S. Xu, "Solving high-dimensional global optimization problems using an improved sine cosine algorithm," *Expert Systems with Applications*, vol. 123, pp. 108–126, 2019.

[50] C. W. Qu, Z. L. Zeng, J. Dai, and W. He, "A modified sine-cosine algorithm based on neighborhood search and Greedy Levy mutation," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 4231267, 19 pages, 2018.

[51] S. N. Chegini, A. Bagheri, and F. Najafi, "PSOSCALF: a new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems," *Applied Soft Computing*, vol. 73, pp. 697–726, 2018.

[52] H. Nenavath and R. K. Jatoth, "Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking," *Applied Soft Computing*, vol. 62, pp. 1019–1043, 2018.

[53] G. B. Dantzing and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[54] J.-Y. Potvin, "State-of-the art review-evolutionary algorithms for vehicle routing," *InCforms Journal on Computing*, vol. 21, no. 4, pp. 518–548, 2009.

[55] Q. Shan and J. C. Wang, "Solve Capacitated Vehicle Routing Problem Using Hybrid Chaotic Particle Swarm Optimization," in *Proceedings of the 2013 Sixth International Symposium on Computational Intelligence and Design Hangzhou*, pp. 422–427, Hangzhou, China, 2013.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1428104, 10 pages
https://doi.org/10.1155/2020/1428104

Research Article

A Short-Term Load Forecasting Method Based on GRU-CNN Hybrid Neural Network Model

Lizhen Wu, Chun Kong, Xiaohong Hao, and Wei Chen

College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, 730050, Gansu, China

Correspondence should be addressed to Lizhen Wu; wulzlut@163.com

Received 19 November 2019; Revised 19 January 2020; Accepted 15 February 2020; Published 21 March 2020

Guest Editor: Chi-Hua Chen

Short-term load forecasting (STLF) plays a very important role in improving the economy and stability of the power system operation. With the smart meters and smart sensors widely deployed in the power system, a large amount of data was generated but not fully utilized, these data are complex and diverse, and most of the STLF methods cannot well handle such a huge, complex, and diverse data. For better accuracy of STLF, a GRU-CNN hybrid neural network model which combines the gated recurrent unit (GRU) and convolutional neural networks (CNN) was proposed; the feature vector of time sequence data is extracted by the GRU module, and the feature vector of other high-dimensional data is extracted by the CNN module. The proposed model was tested in a real-world experiment, and the mean absolute percentage error (MAPE) and the root mean square error (RMSE) of the GRU-CNN model are the lowest among BPNN, GRU, and CNN forecasting methods; the proposed GRU-CNN model can more fully use data and achieve more accurate short-term load forecasting.

1. Introduction

Due to the difficulty of large-scale storage of electrical energy and electrical energy changes in power demand, it is required that the system power generation should be dynamically balanced with changes in load [1, 2]. Load forecasting plays an important role in power construction planning and power grid operation, accurate load forecasting can minimize the gap between electricity supply and demand, improvings the stability of power systems [3]. A tiny little error increased in the load forecasting may cost millions of dollars lost every year [4], and thus it is essential to build an accurate load forecasting model. According to the predicted time range, the power load forecasts can be divided into long-term, medium-term, short-term, and ultra-short-term forecast [5]. This paper mainly focuses on short-term load forecasting, which predicts future loads from minutes to weeks; accurate STLF can help power system staff to develop reasonable production plans, maintain supply and demand balance, and ensure grid safety while reducing resource waste and electricity costs [6, 7].

With the large-scale application of smart meters and smart sensors in the power system, the degree of informatization continues to increase, a large amount of data are generated, and it provides a reliable source of data for accurate load forecasting. Meanwhile, with the continuous improvement of computer computing performance and the application of distributed parallel computing, powerful computing performance for massive data is provided. Under this background, many load forecasting methods based on massive data were emerged, and these methods are mainly divided into two categories, one is traditional statistical methods [8]: they are most frequently used in the early literature, including linear regression (LR) analysis approach and autoregressive moving average (ARMA) approach [9]. Lee C. K. proposed a lifting scheme and autoregressive integrated moving average (ARIMA) models to achieve STLF in [10]. These methods can achieve short-term load forecasting to a certain extent; however, there are many inherent nonlinear features in the massive data, and traditional statistical methods cannot well learn these nonlinear data [11]; so, it is very challenging for these traditional statistical methods to predict accurately in STLF, and these

traditional statistical methods cannot meet the requirements of load forecasting accuracy in the STLF. The other is machine learning methods; they have been widely and successfully used in prediction and classification problems, including artificial neural network (ANN) [12], support vector machine (SVM) [13], and fuzzy inference system (FIS) [14]. For better abstraction of nonlinear features, machine learning methods are good approaches to solve nonlinear problems, and in [15], Niu DX creates a system for power load forecasting using support vector machine and ant colony optimization. In [16], the authors present short-term load forecasting models developed by using the fuzzy logic and adaptive neuro-fuzzy inference system (ANFIS). In recent years, with the rise of artificial intelligence, ANN methods have been widely used in load forecasting. Back-propagation neural network (BPNN) is the first widely used ANN method for STLF [17]. A combined model, which used the back-propagation neural network (BPNN) with the multilabel algorithm based on K-nearest neighbor (K-NN) and K-means, was proposed for STLF in [18]; however, BPNN is a feedforward neural network, and it cannot well learn time sequence data in the power system [19]. In order to efficiently process the time sequence data, such as holiday, weather, and temperature information in the power system.Recurrent neural network (RNN) [20], a kind of neural network which is specific for processing sequence data, is widely used for STLF [21]. The authors in [22] use local RNN models to deal with the problem of long-term wind speed and power forecasting based on meteorological information. However, due to the excessive depth of time and the simple hidden layer in the traditional RNN structure, when the error back propagation is performed, there will be problems with gradient vanishing. It is impossible for RNN to learn long historical data. In response to the shortcomings of RNN, Hochreiter and Schmidhuber proposed a long short-term memory (LSTM) recurrent neural network in 1997 [23], which overcame the disadvantages of traditional RNNs by combining short-term memory with long-term memory through the gate control. A novel method which integrates LSTM and genetic algorithm (GA) was proposed for STLF [24], and it yielded a small mean absolute percentage error. Gated recurrent unit (GRU) [25] is a special type of recurrent neural network based on optimized LSTM, and the GRU internal unit is similar to the internal unit of the LSTM [26], except that the GRU combines the input gate and the forgetting gate in the LSTM into a single update gate. In [27], a novel system called multi-GRU (gated recurrent unit) prediction system was developed based on GRU models for electricity generation's planning and operation. And Wang proposed a novel approach to forecast short-term photovoltaic power based on GRU networks [28]. However, there is not only sequence data in the power system, but also other kinds of high-dimensional data, such as spatiotemporal matrix and image information in the power system. The GRU model cannot well handle all these kinds of high-dimensional data, the convolution neural network (CNN) [29] is ideal for processing high-dimensional data, which has been widely used in image recognition and the fields of prediction [30]. When there is a strong relationship between the nearby data point, CNN can capture local trend features and scale-invariant features [31, 32]. In [33], the author proposed an end-to-end automatic image annotation method based on a deep CNN and multilabel data augmentation, and the model performs well in automatic image annotation.

In order to make full use of the various data in the power system and achieve accurate STLF, the GRU-CNN hybrid neural network model was proposed, which combines the GRU model with the CNN model. In the proposed model, the GRU module is used to model dynamic changes in historical load sequence data for better learning potential features in time sequence data. The CNN module is utilized to process spatiotemporal matrixes and map spatiotemporal matrixes into the feature vector. The GRU-CNN model combines the output of GRU and CNN to derive the load prediction result through the activation function. To verify the superiority of the GRU-CNN model in short-term load forecasting, the proposed method was compared with BPNN, GRU, and CNN models in a real-world experiment. The four models were trained and tested, and mean absolute percentage error (MAPE) and root mean square error (RMSE) were used as the evaluation indexes. The results of the experiments demonstrate that the GRU-CNN model achieves the best predicting performance in STLF among the four models.

This paper is organized as follows: In Section 2, the proposed GRU-CNN hybrid neural networks and its modules are introduced. The GRU-CNN model was utilized to forecast the electrical load, and also it was compared with BPNN, GRU, and CNN in a real-world case in Section 3. Finally, the conclusion is drawn in Section 4.

2. The GRU-CNN Hybrid Neural Network Model

2.1. The Establishment of GRU Module. RNN is a kind of artificial neural network which is suitable for analyzing and processing time sequence data, unlike traditional neural networks, which are based on the weight connection between the layers. RNN applies the hidden layers to preserve information from the previous moment, and the output is influenced by the current states and previous memories. For better understanding of RNN, the unrolled structure of RNN is shown in Figure 1

where $x^{\langle t\rangle}$ and $\hat{y}^{\langle t\rangle}$ represent the input and output at time t, $a^{\langle t\rangle}$ represents the output of one single hidden layer at time t, and $\omega_{aa}^{\langle t\rangle}$, $\omega_{ax}^{\langle t\rangle}$, and $\omega_{ay}^{\langle t\rangle}$ represent the hidden layers weight matrixes, the input weight matrixes, and the output weight matrixes, respectively. Figure 1 can be represented as following formulas:

$$\begin{aligned} a^{<t>} &= g_1\left(\omega_{aa}a^{\langle t-1\rangle} + \omega_{ax}x^{\langle t-1\rangle} + b_a\right), \\ \hat{y}^{<t>} &= g_2\left(\omega_{ay}a^{\langle t\rangle} + b_y\right), \end{aligned} \tag{1}$$

where b_a and b_y represent the bias vectors of one single hidden layer and the output, respectively. and g_1 and g_2 are the nonlinear activation function. RNN performs well when the output is close to its associated inputs; however, when the

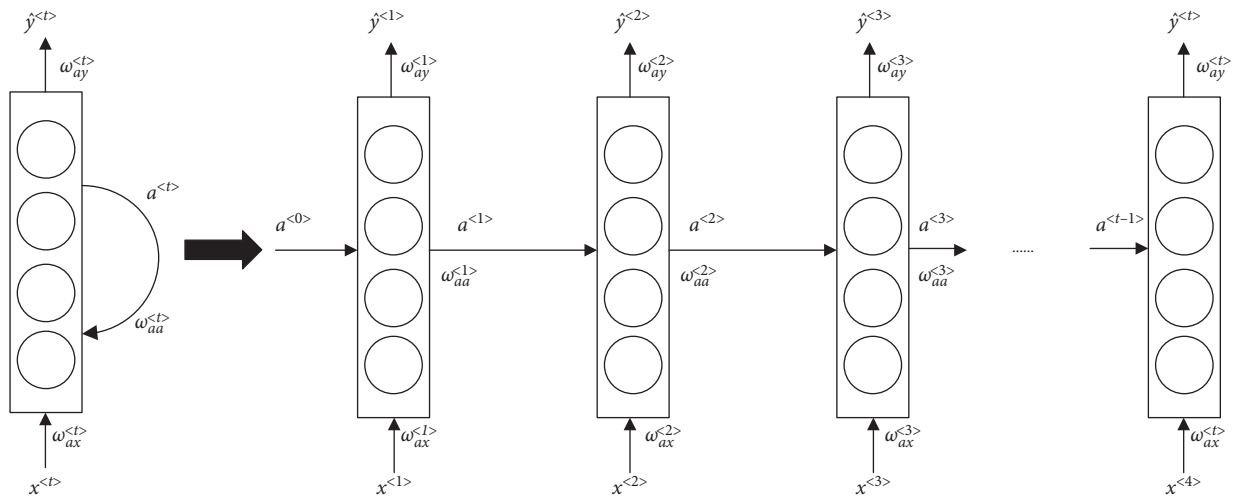


Figure 1: Recurrent neural network structure.

time interval is long and the number of weights becomes large, the input will have little effect on the output due to the gradient vanishing problem. In order to solve the gradient vanishing and simple hidden layer structure problems of RNN, a special type of RNN called GRU was proposed.

GRU is a variant of LSTM with a gated recurrent neural network structure, and comparing with LSTM, there are two gates (update gate and reset gate) in GRU and three gates (forgetting gate, input gate, and output gate) in LSTM; meanwhile, GRU has fewer training parameters than LSTM, so GRU converges quicker than LSTM during training [34]. The GRU structure is shown in Figure 2, where σ and tanh are the activation functions, $c^{\langle t-1\rangle}$ is the input of the current unit, which is also the output of the previous unit, $c^{\langle t\rangle}$ is the output of the current unit, which links to the input of the next unit. $x^{\langle t\rangle}$ are the inputs oftraining data, $\widehat{y}^{\langle t\rangle}$ is the outcome of this unit, generated by the activation function, Γ_r and Γ_u represent the reset gate and the update gate, respectively, and the candidate activation $\widetilde{c}^{\langle t\rangle}$ is computed similarly to that of the traditional recurrent unit. There are two gates in GRU, one is the update gate, which preserve previous information to the current state; The value of Γ_u ranges from 0 to 1, the closer Γ_u is to zero, the more previous information it retains; the other is the reset gate, which is used to determine whether the current status and previous information are to be combined. The value of Γ_r ranges from −1 to 1, the smaller the value of Γ_r, the more previous information it ignores. According to Figure 2, the formulas of GRU can be shown as

$$\begin{aligned}
\Gamma_u &= \sigma\left(\omega_u\left[c^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_u\right),\\
\Gamma_r &= \sigma\left(\omega_r\left[c^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_r\right),\\
\widetilde{c}^{\langle t\rangle} &= \tanh\left(\omega_c\left[\Gamma_r * c^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_c\right),\\
c^{\langle t\rangle} &= \left(1-\Gamma_u\right) * c^{\langle t-1\rangle} + \Gamma_u * \widetilde{c}^{\langle t\rangle},
\end{aligned} \tag{2}$$

where ω_u, ω_r, and ω_c represent the training weight matrix of the update gate, the reset gate, and the candidate activation $\widetilde{c}^{\langle t\rangle}$, respectively and b_u, b_r, and b_c are the bias vectors.

2.2. The Establishment of CNN Module. CNN is a kind of artificial neural network which can well process high-dimensional data. It is commonly applied in visual image, video recognition, and text categorization. There are many smart sensors and devices in the power system. In order to preserve the spatial information of data recorded by smart sensors and devices in the power system, the spatiotemporal matrix was proposed and the spatiotemporal matrixes data are based on the location of the sensors and time sequence. The spatiotemporal matrix is shown as

$$X = \begin{bmatrix} X_1(1) & X_1(2) & \cdots & X_1(n)\\ X_2(1) & X_2(2) & \cdots & X_2(n)\\ \vdots & \vdots & \ddots & \vdots\\ X_k(1) & X_k(2) & \cdots & X_k(n)\end{bmatrix}, \tag{3}$$

where k represents the kth smart sensor, n represents the nth time sequence, and $X_k(n)$ represents the data recorded by the kth smart sensor at n time. In order to extract the load feature from the spatiotemporal matrix, CNN was used to process the spatiotemporal matrix. The structure of CNN is shown in Figure 3.

As shown in Figure 3, firstly, many two-dimensional spatiotemporal matrixes are stacked into three-dimensional matrix blocks, and then these blocks were applied with a convolution operation. The purpose of the convolution operation is to get a highly abstract feature, and after the convolution operation, the outputs of convolution operation are applied with pooling operation. Pooling operation does not change the depth of the input matrix, but it can reduce the size of matrixes and the number of nodes, so as to reduce the parameters in the entire neural networks. After repeated convolution and pooling operations, the highly abstract feature was obtained and flattened to an one-dimensional vector, so it can be connected with the fully connected layer. Then, the weights and bias parameters of the fully connected layer can be calculated iteratively. Finally, prediction results are obtained through the output of activation function.

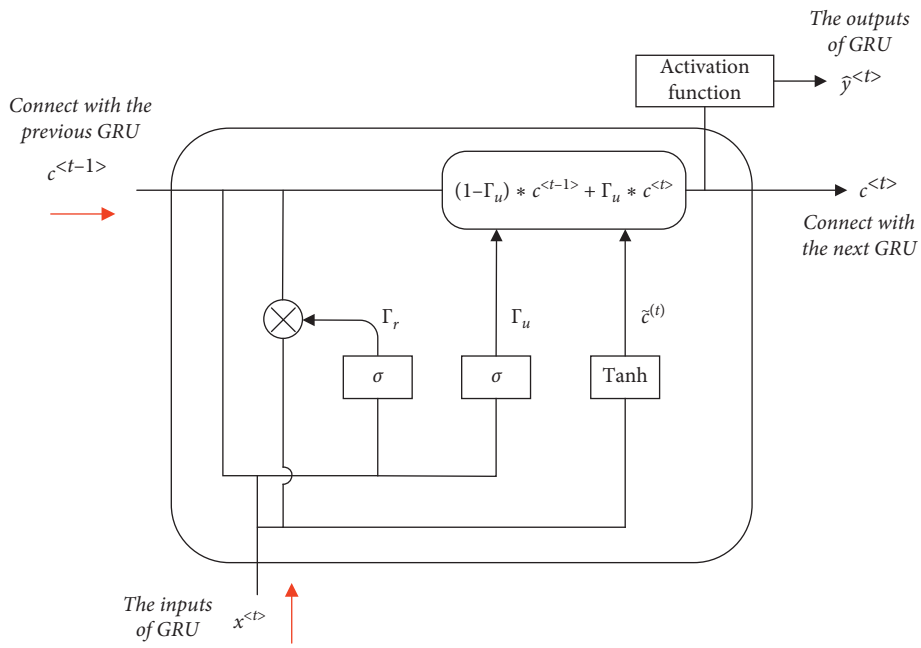


Figure 2: Gated recurrent unit structure.

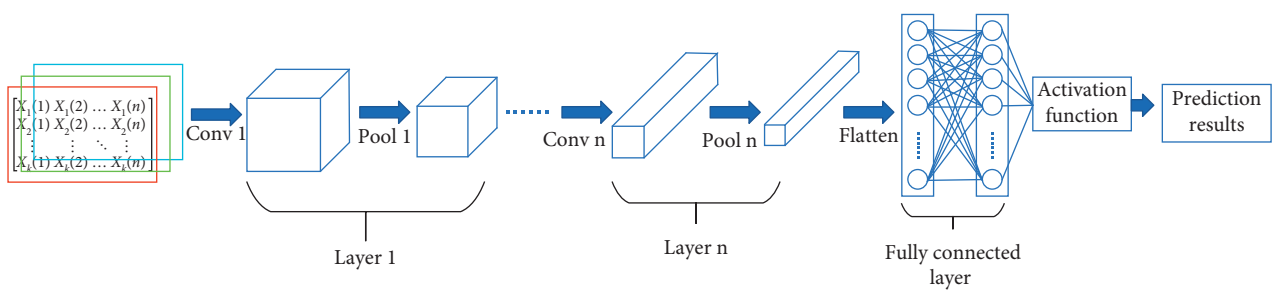


Figure 3: Convolution neural network structure.

2.3. The GRU-CNN Hybrid Neural Networks. To combine the advantages of the GRU module which can well process time sequence data and the advantages of the CNN module which is ideal for handling high-dimensional data, the GRU-CNN hybrid neural networks was proposed, and the structure of GRU-CNN hybrid neural networks is shown in Figure 4.

The framework of the proposed GRU-CNN hybrid neural networks consists of a GRU module and a CNN module. The inputs are the information of time sequence data and spatiotemporal matrixes collected from the power system; the outputs are the prediction of the future load value. In aspect of the CNN module, it is good at processing two-dimensional data, such as spatiotemporal matrixes and images. The CNN module uses local connection and shared weights to directly extract local features from the spatiotemporal matrixes data and obtain effective representation through the convolution layer and pooling layer. The structure of the CNN module contains two convolution layers and a flatten operation, and each convolution layer contains a convolution operation and a pooling operation. After the second pooling operation, the high-dimensional data are flattened into one-dimensional data, and the outputs of the CNN module are connected with the fully connected layer. On the other hand, the aim of the GRU module is to capture the long-term dependency and the GRU module can learn useful information in the historical data for a long period through the memory cell, and the useless information will be forgotten by the forget gate. The inputs of GRU module are time sequence data; the GRU module contains many gated recurrent units, and the outputs of all these gated recurrent units are connected with the fully connected layer. Finally, the load predicting results can be obtained by calculating the mean value of all neurons in the fully connected layers. The flow chart of the GRU-CNN method is shown in Figure 5.

3. Experiments and Results

3.1. Datasets Description. In this experiment, the electric load dataset is provided by a power distribution network in Wuwei, Gansu province. The dataset was collected in August 2018. It contains 44640 samples which were recorded every minute for a total of 31 days. There are 31680 samples that are selected from these datasets as the training set. Also,

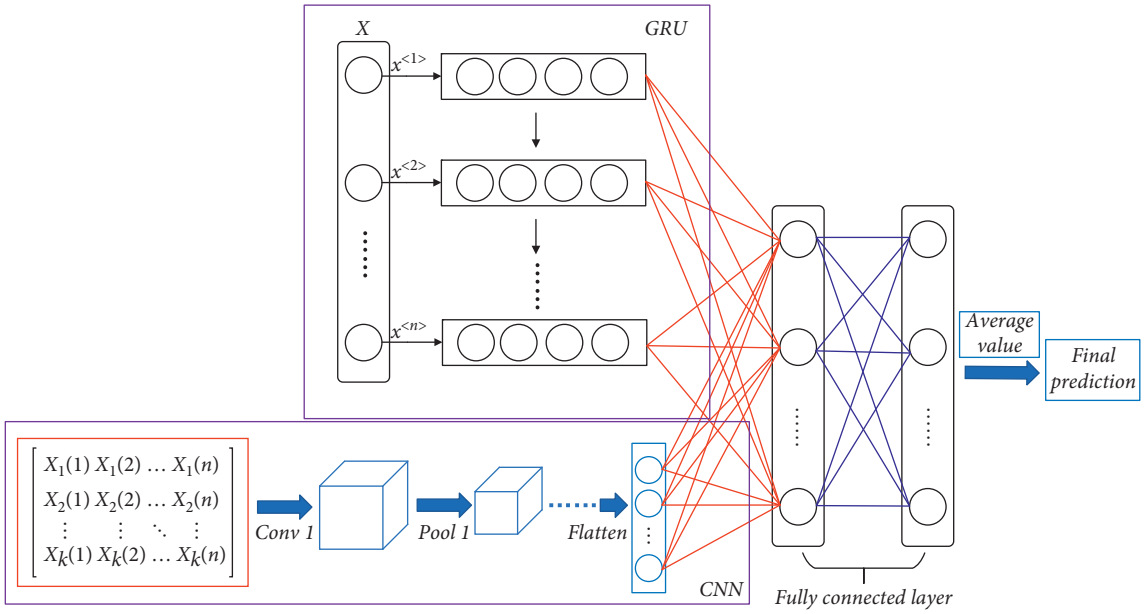


Figure 4: GRU-CNN hybrid neural networks.

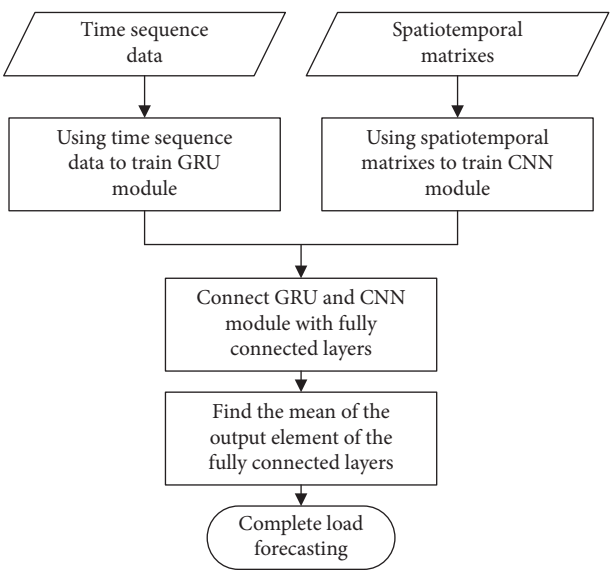


Figure 5: The flow chart of the GRU-CNN method.

11520 samples are utilized as the test set, which is divided into 8 tests, and each test contains 1440 samples. And the last 1440 samples are selected as the dev set. Each sample contains time sequence data such as temperature, holidays, and weather in Wuwei, and spatiotemporal matrixes data, which were collected by distributed smart electric meters in the Wuwei distribution network. Time sequence data of the samples were selected as the input of the GRU module, and spatiotemporal matrixes data of the samples were selected as the input of the CNN module.

3.2. Model Evaluation Indexes. To evaluate the performance of different predicting models, the absolute percentage error (APE), the mean absolute percentage error (MAPE), and the root mean square error (RMSE) are introduced, and the formulas of APE, MAPE, and RMSE are shown as follows:

$$\begin{aligned} \mathrm{APE} &= \left|\frac{y(i)-\hat{y}(i)}{y(i)}\right| \times 100\%, \\ \mathrm{MAPE} &= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y(i)-\hat{y}(i)}{y(i)}\right| \times 100\%, \\ \mathrm{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}|y(i)-\hat{y}(i)|^2}, \end{aligned} \tag{4}$$

where n is the size of the training and test samples, $y(i)$ and $\hat{y}(i)$ are the actual value and the predicted value, respectively. The APE represents the ratio between the error and actual values at one predicting point, the MAPE represents the average of APE in all the test sets, and the RMSE is the sample standard deviation of differences between the predicted value and the actual value; the smaller the values of MAPE and RMSE, the better predicting performance the model achieves.

3.3. Experimental Results and Analyses. In order to evaluate the performance and stability of the GRU-CNN model, several most commonly used neural networks were chosen as contrast models, including back-propagation neural network (BPNN) model, convolutional neural network (CNN) model, and gated recurrent unit (GRU) model. The parameters of BPNN, GRU, CNN, and the GRU-CNN model were obtained by learning the training datasets, and the test datasets were fed to the four forecast models. A way to assess whether a prediction model is well trained and learned is the loss value of the loss function. Cross entropy

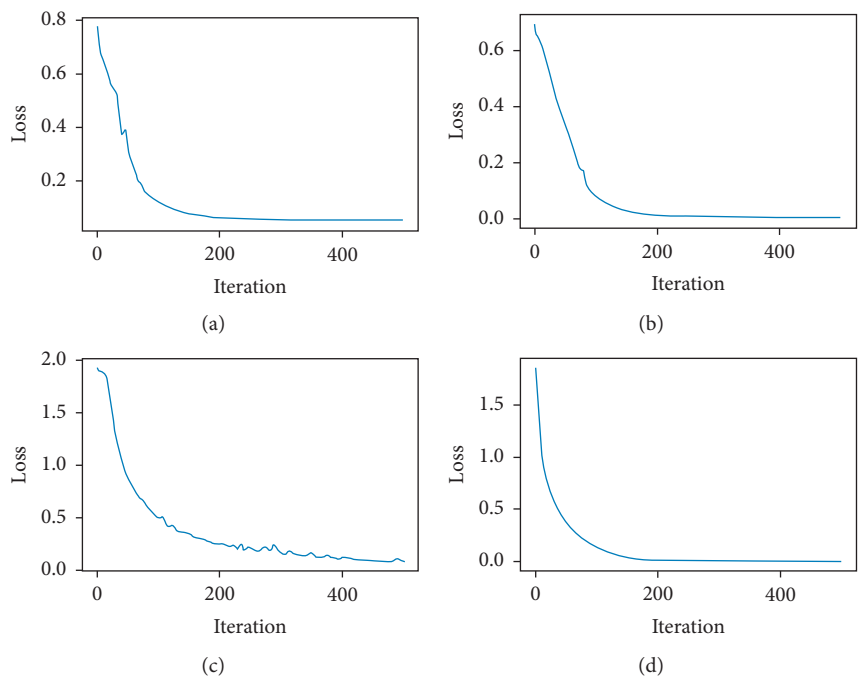


Figure 6: Neural networks' losses on the training set for 500 iterations. (a) BPNN. (b) GRU. (c) CNN. (d) GRU-CNN.

function was selected as loss function, which characterizes the distance between two probability distributions. The smaller the cross entropy, the closer the two probability distributions. The formula of cross entropy is shown as follows:

$$\text{Loss} = -\sum_{i=1}^{n} y_i \log y_i', \tag{5}$$

where n is the size of the training samples and y_i and y_i' are the actual value and the predicted value, respectively. The closer the loss value is to zero, the better the prediction model fits the training set. The loss value of the four load prediction models is shown in Figure 6.

As seen from Figure 6, the BPNN, GRU, and GRU-CNN model converge fast, and the loss values of them are lower than 0.1 after 200 iterations, but the loss value of the CNN model is about 0.2 at the 200th iteration, and there are slight fluctuations in the CNN model. All these methods achieve a very small loss value after 500 iterations, and this means that all the methods can well learn the training set and achieve good performance at the test set. It can be found that training the training set more than 500 iterations, the loss value decreased extremely slowly. And then if they continued to increase the times of iteration, all these models perform well on the test set, but they cannot predict well on the dev set. Obviously, the models were overfitted, and so the times of iteration were settled at 500 times in this experiment.

There are eight groups of data which were trained and tested in the prediction of 24 hours load forecast for the BPNN, GRU, CNN, and GRU-CNN models, and the MAPE and RMSE of the load predicting results of the four models are shown in Table 1 and Table 2, respectively.

Where, test avg represents the average value of the eight testing results. As seen in Tables 1 and 2, the results of the BPNN model performs a little better than the GRU model, while the CNN model performs much better than the GRU model and the BPNN mode. Because most of the datasets in this paper are spatiotemporal matrixes data, there is a certain loss of information for flattening spatiotemporal matrixes data into one-dimensional time sequence data, and so the BPNN and GRU models cannot fully learn the features from the dataset. However, CNN is really good at handling high-dimensional data, and the spatiotemporal matrixes can be processed fully and rapidly. The proposed GRU-CNN model which is the fusion of the GRU model and the CNN model provides the best forecasting results. The test avg indexes of the GRU-CNN model are the minimum among all the four models. The GRU-CNN model can well learn both the time sequence data and the spatiotemporal matrixes and extract more features from the dataset. The average value of MAPE and RMSE of the GRU-CNN model is 2.8839% and 1203.23, respectively. The proposed model has an improved performance of 1.5842% which is better than the BPNN model, 1.7538% better than the GRU model, and 0.5051% better than the CNN model. For better visualization, the MAPE and RMSE lines of the four models are drawn in Figure 7.

As shown in Figure 7, the MAPE and RMSE lines of BPNN and GRU are very close, but the BPNN model performs better than the GRU model in test 3, test 4, test 5, test 6, and test 8. The lines of the CNN model fluctuate between BPNN lines and GRU-CNN lines, and it proves that the CNN model performs always better than the BPNN model and the GRU model but is inferior to the

Table 1: MAPE of the four prediction models.

Test	BPNN (%)	GRU (%)	CNN (%)	GRU-CNN (%)
Test 1	4.1245	4.0132	3.1973	3.1380
Test 2	4.9930	4.8877	3.0290	2.4428
Test 3	4.5197	4.9622	3.1971	2.8844
Test 4	4.1853	4.5516	3.8850	2.8688
Test 5	4.3401	4.7281	3.5980	2.9557
Test 6	4.3920	4.6418	3.5102	3.0947
Test 7	4.7342	4.7136	3.3682	2.8035
Test 8	4.4556	4.6037	3.3271	2.8835
Test avg	4.4681	4.6377	3.3890	2.8839

Table 2: RMSE of the four prediction models.

Test	BPNN	GRU	CNN	GRU-CNN
Test 1	1703.18	1658.33	1329.52	1305.62
Test 2	2053.19	2010.75	1261.69	1025.46
Test 3	1862.45	2040.77	1329.44	1203.43
Test 4	1727.68	1875.30	1606.66	1197.13
Test 5	1790.07	1946.43	1491.00	1232.15
Test 6	1810.98	1911.65	1455.62	1288.17
Test 7	1948.89	1940.59	1398.39	1170.82
Test 8	1836.61	1896.30	1381.83	1203.06
Test avg	1841.63	1910.02	1406.77	1203.23

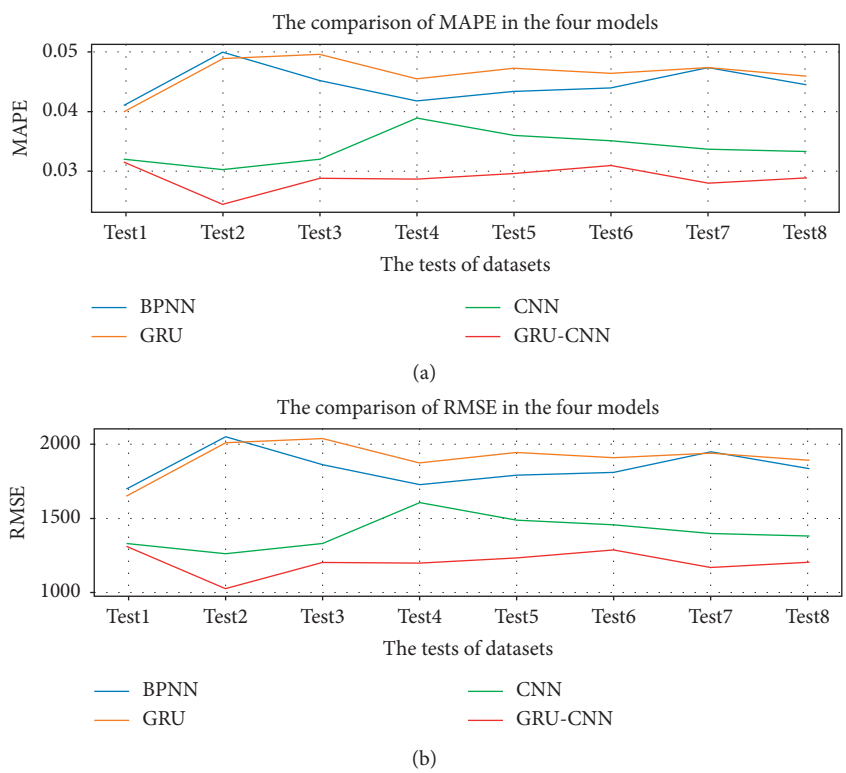


Figure 7: The comparison of MAPE (a) and RMSE (b) of the four models.

GRU-CNN model. The curves of the proposed GRU-CNN model are the minimum among all the eight tests, which demonstrates that the proposed GRU-CNN model achieves the best predicting performance among the four models.

In addition, the dev set, as the inputs of all the four models, was sent to the four trained models to predict the actual load. Finally, the predicting results of the four models were obtained and compared with actual load, as shown in Figure 8.

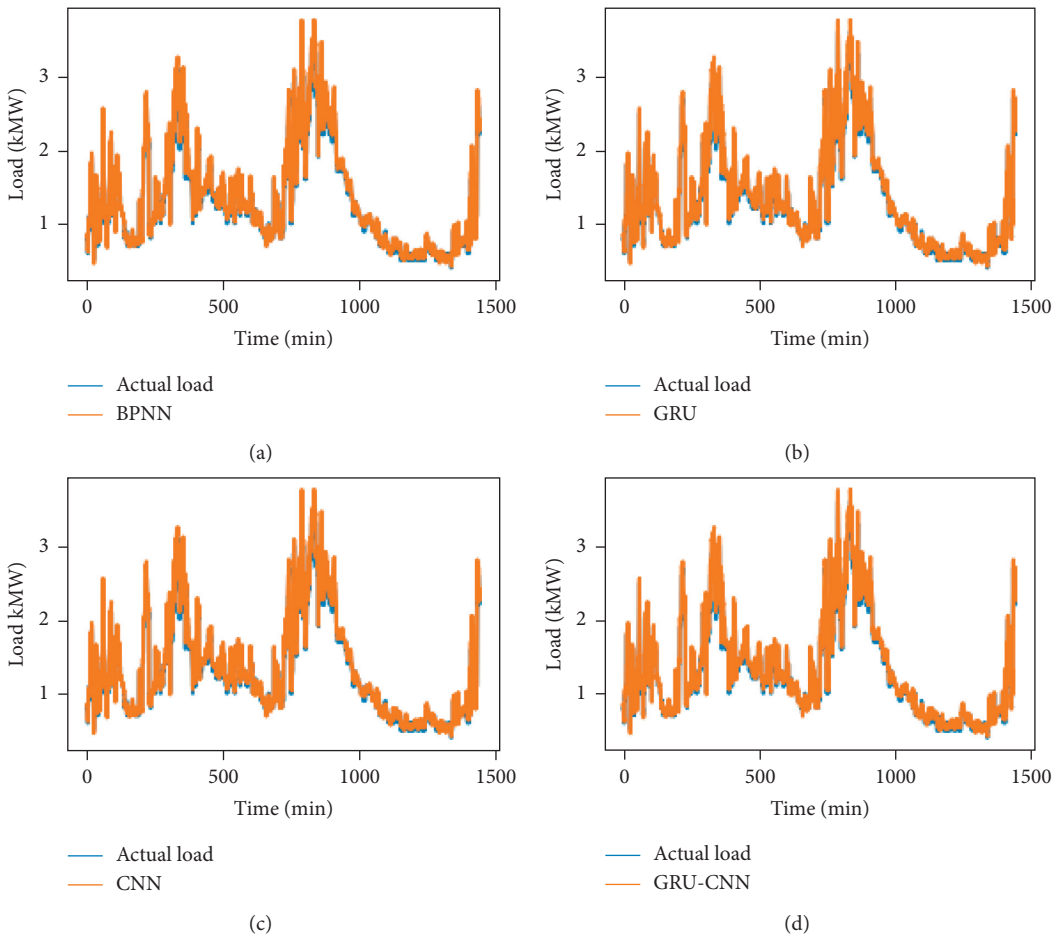


Figure 8: The prediction result of the GRU-CNN model. (a) BPNN. (b) GRU. (c) CNN. (d) GRU-CNN.

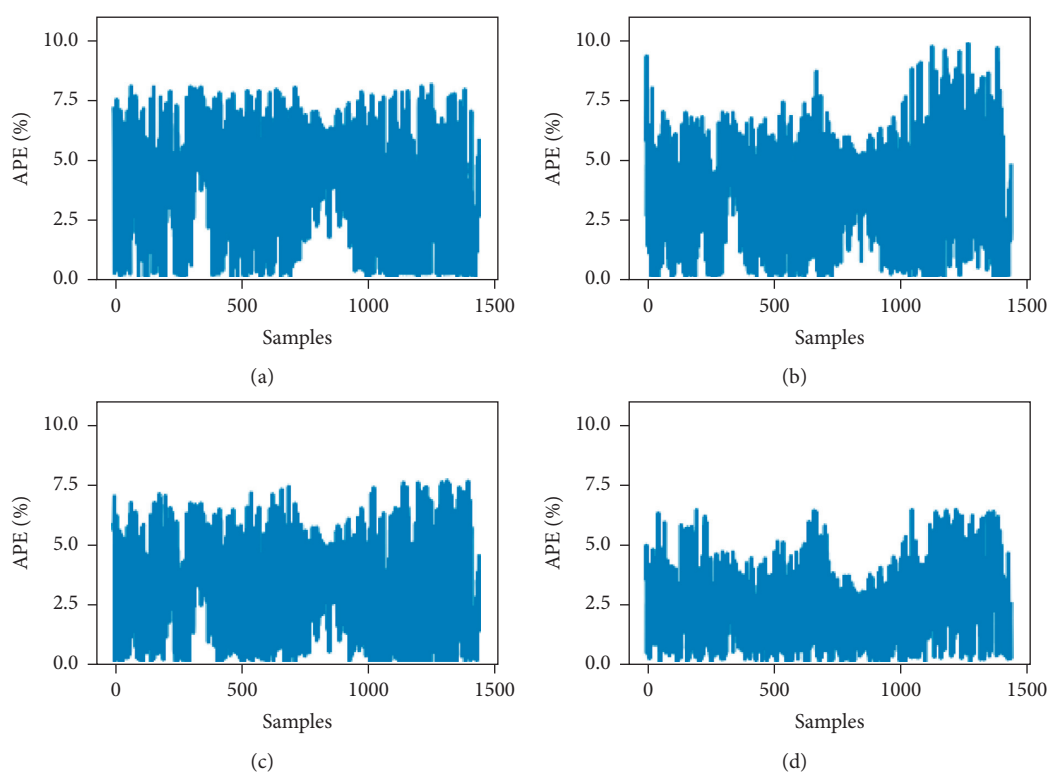


Figure 9: The absolute percentage errors of the GRU-CNN model. (a) The APE of BPNN. (b) The APE of GRU. (c) The APE of CNN. (d) The APE of GRU-CNN.

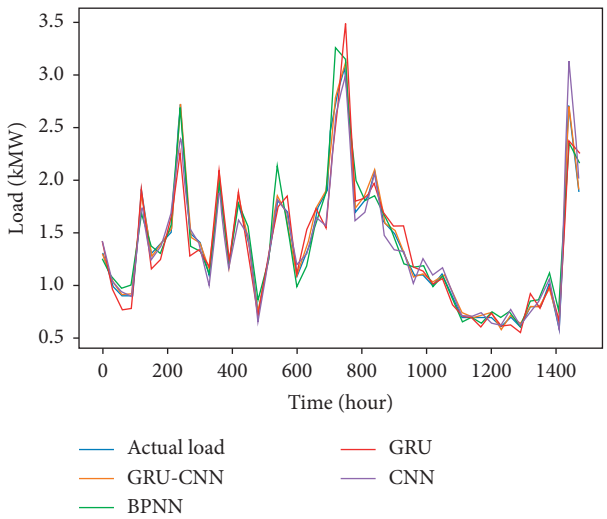


Figure 10: The forecasting load of four models and the actual load.

As shown in Figure 8, the prediction curves of all the four models fit well; whether in the peaks or troughs, the blue and orange lines almost overlap. Due to the large ranges of load changing, it is difficult to see the details of the predicting results, so the absolute percentage errors of the four models are drawn, respectively, in Figure 9.

As can be seen from Figure 9(b), the APE of GRU has the largest fluctuation, and it fluctuates from 0% to 10%, and the max APE of the GRU predicting model is 9.8578%, and the MAPE is 4.8781%. It can be seen from Figure 9(d) that the APE of the GRU-CNN model fluctuates the least from 0% to 7% and the max APE of the GRU-CNN predicting model is 6.418%, and the MAPE is 2.3034%. The max APEs of the BPNN model and the GRU model are 8.1306% and 7.605%, respectively, and the MAPEs are 4.0942% and 3.4581%, respectively. From the experiment of the dev set, the proposed GRU-CNN model achieves the most accurate and stable predicting result. And all the load forecasting results of the four models and the actual load are drawn in Figure 10.

In Figure 10, all the four models can predict well on the dev set, and the proposed GRU-CNN predicting method fits best. When predicting peaks, BPNN, GRU, and CNN perform not very well. BPNN, GRU, and CNN model cannot accurately analyze the load fluctuation law, resulting in a decrease in prediction accuracy. In contrast, the GRU-CNN model can effectively learn the load changing trend and accurately analyze the influence of the input characteristics on the load during the time of reaching the peaks to ensure the prediction accuracy.

4. Conclusions

Aiming at improving the accuracy of STLF, the GRU-CNN hybrid neural network was proposed in this paper, which is based on the GRU model and the CNN model. The GRU module of the GRU-CNN model dedicates to process time sequence data, and the CNN module is good at processing spatiotemporal matrixes. The proposed model can predict electrical load quickly and accurately by extracting features from variable data that affect the power system. In the real-world experiments of the Wuwei area electrical load forecasting, the proposed GRU-CNN model was compared with BPNN, GRU, and CNN models. The results show the proposed GRU-CNN model can well process both time sequence data and spatiotemporal matrixes data and can effectively extract the hidden feature of the datasets. The GRU-CNN model has the lowest value of MAPE and RMSE, and it demonstrates that the proposed GRU-CNN model achieves the best performance among all the four models.

In further study, the proposed model can be improved by collecting and adding more relevant factors that may affect the load change. At the same time, in order to train the neural network more effectively, a new loss function training method is designed to reduce the training time of the model. It can even be considered for random load forecasting, such as electric vehicles.

Data Availability

The xls data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This project was supported by the National Natural Science Foundation of China (no. 51467009); the Basic Research Innovation Group Project of Gansu Province (no. 18JR3RA133).

References

[1] B. Yildiz, J. I. Bilbao, and A. B. Sproul, "A review and analysis of regression and machine learning models on commercial building electricity load forecasting," *Renewable and Sustainable Energy Reviews*, vol. 73, pp. 1104–1122, 2017.

[2] K. Ni, J. Wang, G. Tang, and D. Wei, "Research and application of a novel hybrid model based on a deep neural network for electricity load forecasting: a case study in Australia," *Energies*, vol. 12, no. 13, p. 2467, 2019.

[3] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019.

[4] L. Yang and H. Yang, "Analysis of different neural networks and a new architecture for short-term load forecasting," *Energies*, vol. 12, no. 8, 2019.

[5] C. Tian, J. Ma, C. Zhang, and P. Zhan, "A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network," *Energies*, vol. 11, no. 12, 2018.

[6] J. Zhang, Y.-M. Wei, D. Li, Z. Tan, and J. Zhou, "Short term electricity load forecasting using a hybrid model," *Energy*, vol. 158, pp. 774–781, 2018.

[7] N. Ghadimi, A. Akbarimajd, H. Shayeghi, and O. Abedinia, "Two stage forecast engine with feature selection technique

and improved meta-heuristic algorithm for electricity load forecasting," *Energy*, vol. 161, pp. 130–142, 2018.

[8] A. Goia, C. May, and G. Fusai, "Functional clustering and linear regression for peak load forecasting," *International Journal of Forecasting*, vol. 26, no. 4, pp. 700–711, 2010.

[9] A. H. Nury, K. Hasan, and M. J. B. Alam, "Comparative study of wavelet-ARIMA and wavelet-ANN models for temperature time series data in northeastern Bangladesh," *Journal of King Saud University-Science*, vol. 29, no. 1, pp. 47–61, 2017.

[10] C.-M. Lee and C.-N. Ko, "Short-term load forecasting using lifting scheme and ARIMA models," *Expert Systems With Applications*, vol. 38, no. 5, pp. 5902–5911, 2011.

[11] G.-Y. Chen, M. Gan, and G.-L. Chen, "Generalized exponential autoregressive models for nonlinear time series: stationarity, estimation and applications," *Information Sciences*, vol. 438, pp. 46–57, 2018.

[12] L. Hernández, C. Baladrón, J. Aguiar et al., "Artificial neural network for short-term load forecasting in distribution systems," *Energies*, vol. 7, no. 3, pp. 1576–1598, 2014.

[13] J.-p. Liu and C.-l. Li, "The short-term power load forecasting based on sperm whale algorithm and wavelet least square support vector machine with DWT-IR for feature selection," *Sustainability*, vol. 9, no. 7, 2017.

[14] A. Jadidi, R. Menezes, N. de Souza, and A. C. de Castro Lima, "Short-term electric power demand forecasting using NSGA II-anfis model," *Energies*, vol. 12, no. 10, p. 1891, 2019.

[15] D. Niu, Y. Wang, and D. D. Wu, "Power load forecasting using support vector machine and ant colony optimization," *Expert Systems With Applications*, vol. 37, no. 3, pp. 2531–2539, 2010.

[16] H. H. Çevik and M. Çunkaş, "Short-term load forecasting using fuzzy logic and ANFIS," *Neural Computing and Applications*, vol. 26, no. 6, pp. 1355–1367, 2015.

[17] Z. Xiao, S.-J. Ye, B. Zhong, and C.-X. Sun, "BP neural network with rough set for short term load forecasting," *Expert Systems With Applications*, vol. 36, no. 1, pp. 273–279, 2009.

[18] X. Sun, Z. Ouyang, and D. Yue, "Short-term load forecasting model based on multi-label and BPNN," *Communications in Computer and Information Science*, Springer, New york, NY, USA, pp. 263–272, 2017.

[19] X. Xiong, P. Zhou, and A. Chen, "Asymptotic Normality of the Local Linear Estimation of the Conditional Density for Functional Time Series Data," *Communication In Statistics-Theory And Methods*, vol. 47, no. 14, pp. 3418–3440, 2017.

[20] G. Li, H. Wang, S. Zhang, J. Xin, and H. Liu, "Recurrent neural networks based photovoltaic power forecasting approach," *Energies*, vol. 12, no. 13, 2019.

[21] D. Yi, S. Bu, and I. Kim, "An enhanced algorithm of RNN using trend in time-series," *Symmetry*, vol. 11, no. 7, p. 912, 2019.

[22] T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis, and P. S. Dokopoulos, "Long-term wind speed and power forecasting using local recurrent neural network models," *IEEE Transactions on Energy Conversion*, vol. 21, no. 1, pp. 273–284, 2006.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] A. S. Santra and J.-L. Lin, "Integrating long short-term memory and genetic algorithm for short-term load forecasting," *Energies*, vol. 12, no. 11, 2019.

[25] J. Chung, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *Clinical Orthopaedics and Related Research*, vol. 3555, 2014.

[26] H. Cheng, Z. Xie, L. Wu, Z. Yu, and R. Li, "Data prediction model in wireless sensor networks based on bidirectional LSTM," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, Article ID 203(2019), 2019.

[27] W. Li, T. Logenthiran, and W. L. Woo, "Multi-GRU prediction system for electricity generation's planning and operation," *IET Generation Transmission & Distribution*, vol. 13, no. 9, pp. 1630–1637, 2019.

[28] Y. Wang, W. Liao, and Y. Chang, "Gated recurrent unit network-based short-term photovoltaic forecasting," *Energies*, vol. 11, no. 8, 2018.

[29] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[30] W. Liu, Y. Song, D. Chen et al., "Deformable object tracking with gated fusion," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3766–3777, 2019.

[31] P.-H. Kuo and C.-J. Huang, "A high precision artificial neural networks model for short-term energy load forecasting," *Energies*, vol. 11, no. 1, 2018.

[32] X. Ke, L. Shi, W. Guo, and D. Chen, "Multi-dimensional traffic congestion detection based on fusion of visual features and convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2157–2170, 2019.

[33] X. Ke, J. Zou, and Y. Niu, "End-to-End automatic image annotation based on deep CNN and multi-label data augmentation," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2093–2106, 2019.

[34] X. Gao, X. Li, B. Zhao, W. Ji, X. Jing, and Y. He, "Short-term electricity load forecasting model based on EMD-GRU with feature selection," *Energies*, vol. 12, no. 6, 2019.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 3739040, 12 pages
https://doi.org/10.1155/2020/3739040

Research Article

The Fusion of Unmatched Infrared and Visible Images Based on Generative Adversarial Networks

Yuqing Zhao, Guangyuan Fu, Hongqiao Wang, and Shaolei Zhang

Xi'an Research Institute of High-Tech, Shaanxi 710025, China

Correspondence should be addressed to Yuqing Zhao; zoe_rabbi@126.com

Received 27 December 2019; Accepted 27 February 2020; Published 20 March 2020

Guest Editor: Fangying Song

Visible images contain clear texture information and high spatial resolution but are unreliable under nighttime or ambient occlusion conditions. Infrared images can display target thermal radiation information under day, night, alternative weather, and ambient occlusion conditions. However, infrared images often lack good contour and texture information. Therefore, an increasing number of researchers are fusing visible and infrared images to obtain more information from them, which requires two completely matched images. However, it is difficult to obtain perfectly matched visible and infrared images in practice. In view of the above issues, we propose a new network model based on generative adversarial networks (GANs) to fuse unmatched infrared and visible images. Our method generates the corresponding infrared image from a visible image and fuses the two images together to obtain more information. The effectiveness of the proposed method is verified qualitatively and quantitatively through experimentation on public datasets. In addition, the generated fused images of the proposed method contain more abundant texture and thermal radiation information than other methods.

1. Introduction

Image fusion involves the use of mathematical methods to comprehensively process important information acquired by multiple sensors to produce a composite image that is easier to understand, thereby greatly improving the utilization rate of the image information and the reliability and automation degree of systems for target detection and recognition. Image fusion technology currently plays an important role in military, remote sensing, medicine, computer vision, target recognition, intelligence acquisition, and other applications. The fusion of visible and infrared images is one of the most useful cases of applying this technology. Infrared imaging sensors capture the heat radiation emitted by objects. On the one hand, infrared images are less affected by dark or severe weather conditions but typically lack sufficient background and contour edge details. On the other hand, visible images obtained by spectral reflection offer high resolution, excellent image quality, and rich background details but cannot detect objects under hidden or low light and night conditions. The advantages of visible and infrared images can be combined by constructing fused images that retain richer feature information, making them suitable for subsequent processing tasks.

Image fusion is divided into three levels (from lowest to highest level): pixel-, feature-, and decision-level image fusion. Currently, the most highly studied and frequently applied image fusion is performed at the pixel level, and the majority of proposed image fusion algorithms work at this level. According to different image fusion processing domains, image fusion can be roughly divided into two categories: the spatial domain and the transform domain. Image fusion based on the spatial domain is directly conducted on the pixel gray space of an image. Common image fusion methods based on the spatial domain include linear weighted image fusion, false color image fusion, image fusion based on modulation, image fusion based on statistics, and image fusion based on neural networks [1, 2]. Image fusion based on the transformation domain involves transforming multisource images, combining coefficients from the transformation to obtain transformation coefficients of the fused images, and conducting inverse

transformation to obtain the fused images. Common fusion algorithms based on the transform domain include those based on the discrete cosine transform (DCT), the fast Fourier transform (FFT), the multiscale transform [3–5], image subspace technology [6, 7], the saliency method [8, 9], the sparse representation method [10, 11], and others [12–15].

Currently, image fusion based on the transform domain is a widely researched method. Most image fusion algorithms are based on multiscale decomposition and typically use the same transformation or representation for different source images. Since the thermal radiation in infrared images and the texture information in visible images are different, multiscale decomposition methods are not suitable for the fusion of infrared and visible images. To overcome this problem, the developers of FusionGAN [16] proposed an infrared and visible image fusion method based on the novel perspective of generative adversarial networks (GANs) [17]. In FusionGAN, the visible image and the generated fused image are both allowed to enter the discriminator. To "deceive" the discriminator, the fused image retains more visible information and drops its infrared thermal radiation information. It is often difficult to obtain perfectly matched infrared and visible images. To solve the above problems, this paper proposes a new fusion method that generates matching infrared images for visible images and produces fusion images that retain visible light texture details and infrared thermal radiation information, as shown in Figure 1.

The main contributions of this paper are as follows:

A new GAN framework is proposed to retain more information from visible and infrared images in fused images.

For visible images without matching infrared images, approximate infrared images are generated to facilitate the next image fusion.

To verify the feasibility and effectiveness of the proposed method, experiments are conducted on publicly available visible and infrared image datasets, and a comparison of the proposed method with other methods is carried out using several popular evaluation methods.

The remainder of the paper is arranged as follows: Section 2 briefly describes related studies on image fusion and GANs. Section 3 introduces the proposed method. In Section 4, the fusion performance of the proposed method is experimentally evaluated. Section 5 presents the conclusion of the paper.

2. Related Studies

In this section, several methods for the fusion of visible and infrared images are briefly introduced along with GANs.

2.1. Infrared and Visible Image Fusion Using a Deep Learning Framework. A method to generate fusion images of infrared and visible images using a deep learning framework was proposed by Li et al. [18]. The authors decompose the source image into base and detail content. The base content is fused by weight averaging. A deep learning network is used to extract multilayer features, and then, the $L1$-norm and a weighted average strategy are used to generate candidates for the fused detail content. The final fused detail content is obtained using a max selection strategy.

2.2. Infrared and Visual Image Fusion through Infrared Feature Extraction and Visual Information Preservation. Zhang et al. [19] proposed an image fusion method that uses infrared feature extraction and visual information preservation. This method uses quadtree decomposition and Bézier interpolation to reconstruct the infrared background and then subtracts the reconstructed background from the infrared image to obtain infrared bright features. The processed infrared features are then added to the visible images to achieve the final fusion image.

2.3. GAN and Its Derivatives

2.3.1. GAN. A GAN [17] consists of a generator G and a discriminator D that perform a minimax game together. The generator attempts to generate realistic images to trick the discriminator, and the discriminator must distinguish all the real images and the images generated by the generator until G and D reach the Nash equilibrium:

$$\min_{G}\max_{D}\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)]+\mathbb{E}_{z\sim p_{z}(z)}[\log(1-D(G(z)))]. \tag{1}$$

2.3.2. Conversion between Matched Images. The Pix2Pix [20] model, based on a conditional GAN (CGAN) [21], realizes the translation task of a variety of matched images. In Pix2Pix, generator G does not require random noise and only accepts one input image X as condition C with translated image Y as the output. Meanwhile, discriminator D accepts an X sample and a Y sample, where Y contains the real sample and the sample generated by the generator and D determines whether X and Y are the actual matched images.

2.3.3. Fusion of Visible and Infrared Images. Recently, Ma et al. proposed FusionGAN [16], which uses a GAN to fuse the thermal radiation information of infrared images with the high resolution and clear texture details of visible images. FusionGAN's generator produces a fused image with infrared intensity and an additional visible gradient, and the discriminator distinguishes the fused image from the real visible image so that the fused image retains both infrared and visible image information.

3. Method

This section introduces the method proposed in this paper. First, the structural framework of the model is described, and then the model is described in greater detail.

Figure 1: Generated fused images.

3.1. Structural Framework of the Model. In this paper, the GAN two-player game is used to fuse visible and infrared images. The structural framework of the training process is shown in Figure 2. Visible image I_V is input as condition C into generator G_1 to generate a fake infrared image I_{RG}. Next, the visible image and the fake infrared image are input into generator G_2 together in a concatenated channel, creating fused image I_F as the output. Discriminator D_1 distinguishes between real visible image I_V and fused image I_F so that the fused image is closer to the visible image and has more visible texture details. Simultaneously, discriminator D_2 distinguishes real infrared image I_R, generated infrared image I_{RG}, and fused image I_F. Through continuous updating, the generated infrared image becomes closer to the real infrared image, and the fused image contains more thermal radiation information. In Figure 3, a visible image is input to obtain a fused image with both visible and infrared radiation information.

3.2. Network Structure. In our model, generator G_1 is the three-part convolutional neural network (CNN), as shown in Figure 4. It contains a downsampling component for convolution, an upsampling component for deconvolution, and a tanh active component. The downsampling component for convolution contains 7 convolution blocks. Except for the first block, each block contains one convolution layer and one LeakReLU active layer. The upsampling component for convolution also contains 7 convolution blocks. The convolution layer adopts a 4×4 filter, with a step size of 2 and "same" padding.

Generator G_2 is the simple five-layer CNN, as shown in Figure 5. The first two layers use a 5×5 filter, layers 3 and 4 use a 3×3 filter, and the last layer uses a 1×1 filter. Each convolution layer has a step size of 1 without padding.

Discriminators D_1 and D_2 adopt the network structure, as shown in Figure 6. The discriminator contains a four-layered CNN and a linear, fully connected layer. The first four convolution layers use a 3×3 filter with a step size of 2, no padding, and the LeakReLU active layer. The last fully connected layer is used for classification.

3.3. Loss Function. The loss function of the proposed method consists of four elements, i.e., loss function L_{G_1} of generator G_1, loss function L_{G_2} of generator G_2, loss function L_{D_1} of discriminator D_1, and loss function L_{D_2} of discriminator D_2. The loss function L_{G_1} of generator G_1 is given by equation (2), where the first term represents adversarial loss between the generator and discriminator D_2 and the second term represents the loss of the structural similarity between the input visible image and the output infrared image:

$$\begin{aligned}\mathscr{L}_{G_1} &= \frac{1}{N}\sum_{n=1}^{N}\left[\log\left(1-D\left(G_1\left(I_V^n,z\right)\right)\right)\right]\\ &\quad+\frac{1}{N}\sum_{n=1}^{N}\left[\left\|I_G^n-G_1\left(I_V^n,z\right)\right\|_1\right].\end{aligned}\tag{2}$$

The loss function L_{G_2} of generator G_2 contains the adversarial losses between generator G_2 and discriminators D_1 and D_2 and the content loss of the fused image relative to the visible and infrared images:

$$\begin{aligned}\mathscr{L}_{G_2} &= \frac{1}{N}\sum_{n=1}^{N}\left(D_1\left(I_F^n\right)-c_1\right)^2+\frac{1}{N}\sum_{n=1}^{N}\left(D_2\left(I_F^n\right)-c_2\right)^2\\ &\quad+\frac{\lambda}{HW}\left(\left\|I_F-I_R\right\|_F^2+\xi\left\|\nabla I_F-\nabla I_V\right\|_F^2\right).\end{aligned}\tag{3}$$

The loss function L_{D_1} of discriminator D_1 is defined as follows:

$$\mathscr{L}_{D_1}=\frac{1}{N}\sum_{n=1}^{N}\left(D_1\left(I_V^n\right)-b_1\right)^2+\frac{1}{N}\sum_{n=1}^{N}\left(D_1\left(I_F^n\right)-a_1\right)^2.\tag{4}$$

The first term represents the classification results of the visible images, and the second term represents those of the fused images. The loss function L_{D_2} of discriminator D_2 is given by equation (5), which includes an additional term to represent the classification results of the generated infrared images:

$$\begin{aligned}\mathscr{L}_{D2} &= \frac{1}{N}\sum_{n=1}^{N}\left(D_2\left(I_R^n\right)-b_2\right)^2+\frac{1}{N}\sum_{n=1}^{N}\left(D_2\left(I_F^n\right)-a_2\right)^2\\ &\quad+\frac{1}{N}\sum_{n=1}^{N}\left(D_2\left(I_G^n\right)-d_2\right)^2.\end{aligned}\tag{5}$$

Figure 2: Framework of the proposed method's training process.

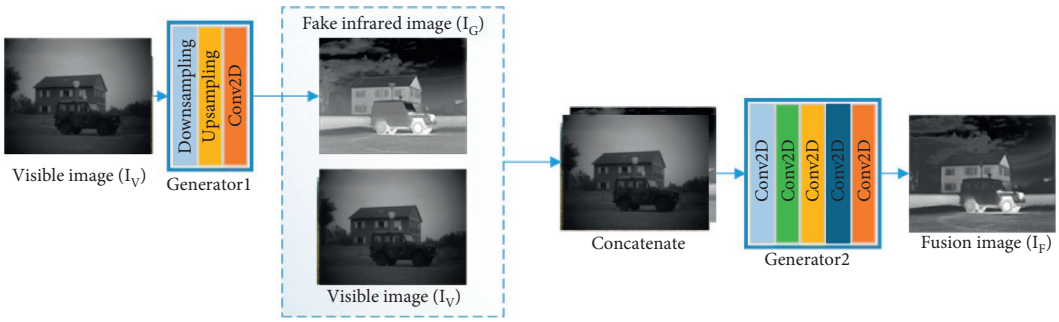


Figure 3: Test procedure of the proposed method.

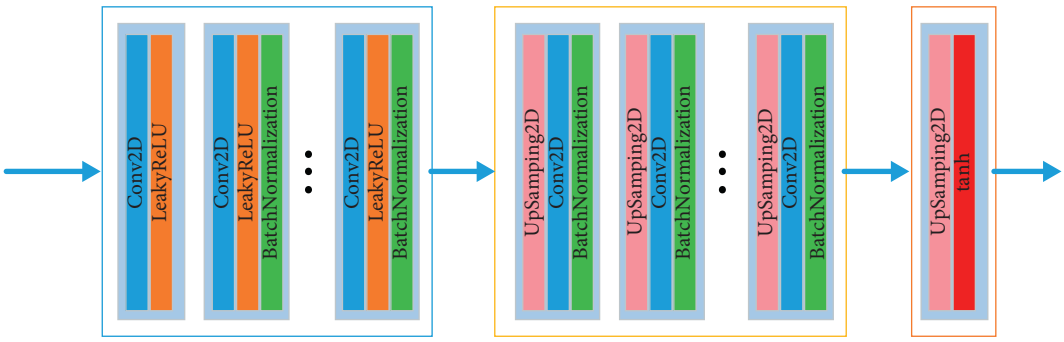


Figure 4: Network structure of generator G_1.

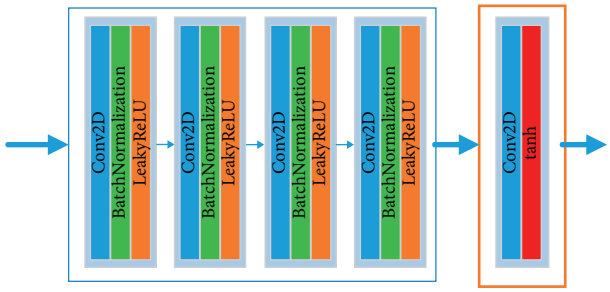


Figure 5: Network structure of generator G_2.

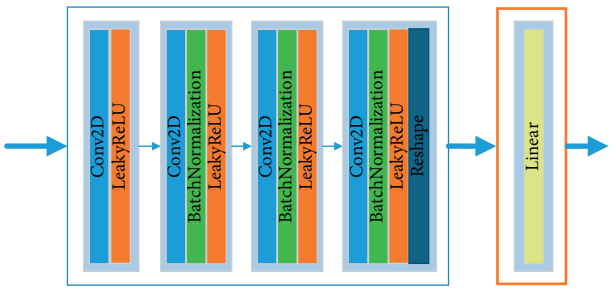


Figure 6: Network structure of the discriminator.

4. Experiments

The TNO Image Fusion Dataset, a commonly used infrared and visible fusion image dataset containing night-vision infrared images and visible images of different scenes, was used to train the proposed method. A total of 56 pairs of images from the TNO dataset were selected. After the images were translated, zoomed, and flipped, 25,936 pairs of infrared and visible images were obtained. All the experiments were performed on a desktop computer with a 2.20 GHz × 40 Intel Xeon(R) Silver 4114 CPU, GeForce GTX 1080 Ti GPU, and 64 GB of internal memory. The training parameters were set to an image batch size of 32 and a learning rate of 10^{-4}, and the generator was trained once for every 2 discriminator training runs. The chosen optimizer was Adam. It took 16.5 hours to train the model. In the first part of this section, several common image fusion evaluation indexes are introduced. In the second part, two datasets are used to validate the effectiveness of the proposed method compared with three popular image fusion methods.

4.1. Common Image Fusion Evaluation Indexes. The evaluation of fused images is performed by combining multiple indexes together. Objective quantitative evaluation methods are mainly divided into two categories: nonreference and reference image evaluation methods. Nonreference image evaluation methods include standard deviation (SD) [22] and information entropy (EN) [23]. Reference image evaluation methods include the correlation coefficient (CC) [24], peak signal-to-noise ratio (PSNR) [25], structural similarity index measure (SSIM) [25], visual information fidelity (VIF) [26], root mean square error (RMSE), and universal image quality index (UIQI) [27] methods. These indexes are defined as follows.

SD reflects the dispersion of the relative mean gray value and is mathematically defined as follows:

$$\mathrm{SD} = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(F(i,j)-\mu)^2}, \tag{6}$$

where μ is the average value of the fused images ($M \times N$). A greater SD value indicates a higher contrast in the fused image and a typically better visual effect.

EN is a statistical feature form that reflects the average amount of information in an image. EN is mathematically defined as follows:

$$\mathrm{EN} = -\sum_{l=0}^{L-1} p_l \log_2 p_l, \tag{7}$$

where L represents the image's gray levels and p_l represents the proportion of pixels with gray value i in the total pixels. A larger EN means that a greater amount of information exists in the fused images.

The CC measures the degree of linear correlation between a fused image and infrared and visible images and is mathematically defined as follows:

$$\mathrm{CC}(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}, \tag{8}$$

where $\mathrm{Cov}(X,Y)$ represents the covariance of X and Y and Var (X) and Var (Y) represent the variance of X and Y, respectively. The larger the CC is, the higher the degree of correlation between the fused images and visible and infrared images is, and the higher the similarity is.

The PSNR assumes that the difference between a fused and reference image is noise and is mathematically defined as follows:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right), \tag{9}$$

where MAX represents the maximum value of the image color and MSE is the mean squared error. The larger the PSNR is, the more similar the two images are. The common benchmark is 30 dB, and fused images with PSNR < 30 dB are clearly deteriorated.

The SSIM considers image distortion by comparing changes in image structure information, thereby obtaining an objective quality evaluation. The mathematical definition of SSIM is as follows:

$$\mathrm{SSIM}(X,Y) = \left(\frac{2u_x u_y + c_1}{u_x^2 + u_x^2 + c_1}\right)^{\alpha} \cdot \left(\frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}\right)^{\beta} \cdot \left(\frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}\right)^{\gamma},$$
$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y), \tag{10}$$

where x and y are the reference image and fused image, respectively, u_x, u_y, σ_x^2, σ_y^2, and σ_{xy} represent the mean value and variance and covariance of images x and y, respectively, and c_1, c_2, and c_3 are small normal numbers to avoid having a

Figure 7: Qualitative fusion results of six images from the TNO dataset: (a) visible images; (b) infrared images; (c) DenseFuse fusion results; (d) DeepFuse fusion results; (e) FusionGAN fusion results; (f) proposed method fusion results.

denominator of zero. Parameters α, β, and γ are used to adjust the proportions.

VIF is a reference image evaluation method based on natural scene statistics and the concept of image information extracted from the human visual system. The mathematical definition of VIF is as follows:

$$\mathrm{VIF} = \frac{\sum_{k=1}^{K} I\left[\left(C_r^k; F^k \mid z_r^k\right)\right]}{\sum_{k=1}^{K} I\left[\left(C_r^k; E^k \mid z_r^k\right)\right]}, \tag{11}$$

where $I(C; E \mid z)$ is the reference image information content and $I(C; F \mid z)$ is the mutual information of the reference and fused images.

The RMSE is defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[R(i,j) - F(i,j)^2\right]}. \tag{12}$$

Figure 8: Continued.

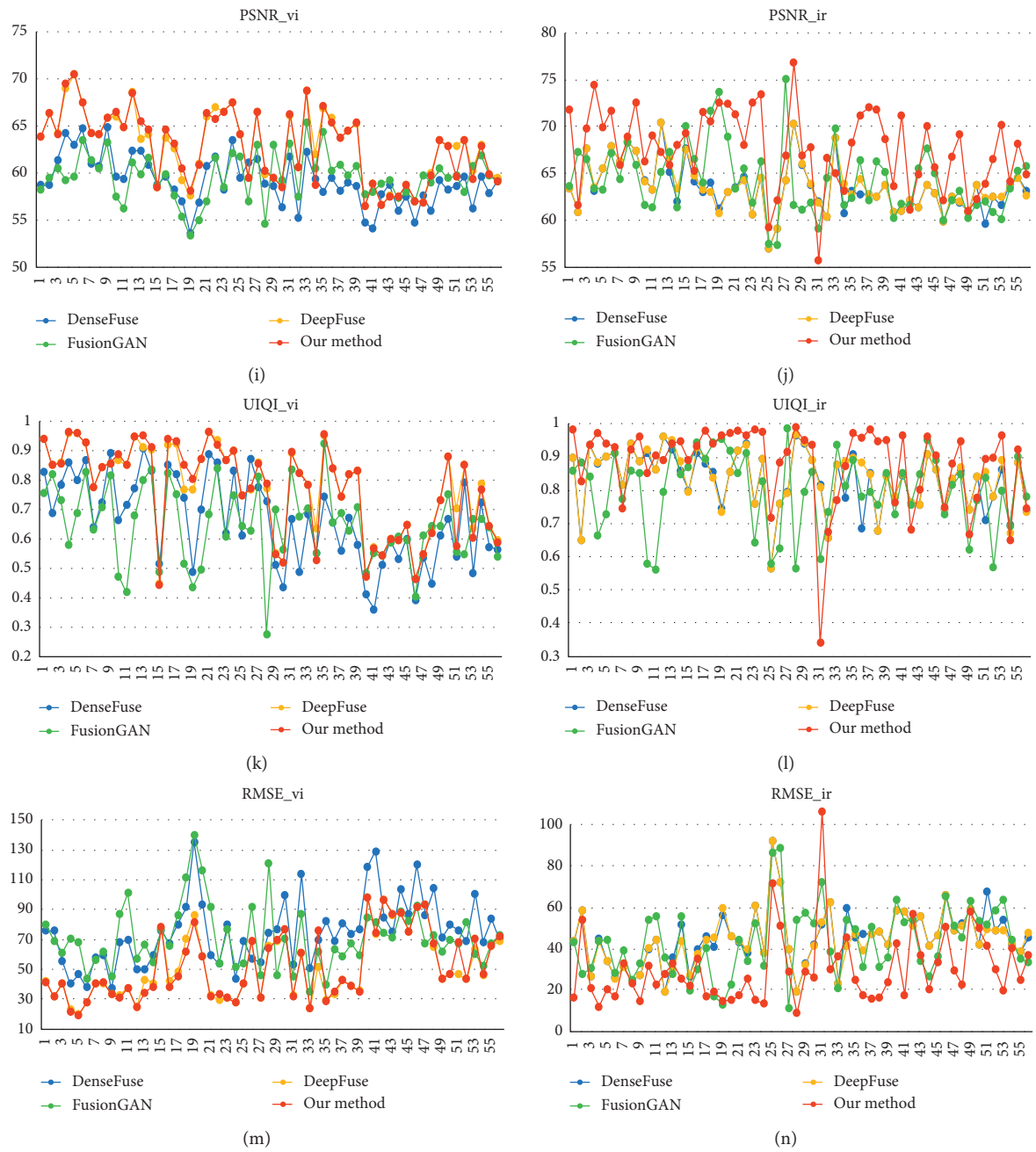


Figure 8: Quantitative comparison of the 14 dimensions of 8 indexes for 56 pairs of images from the TNO dataset.

The UIQI measures image distortion via a combination of three factors: loss of correlation, brightness distortion, and contrast distortion.

4.2. Experimental Validation on Fusion Performance

4.2.1. Validation by the TNO Dataset. Qualitative comparison: to provide a more intuitive observation of fusion performance, six representative images were selected for qualitative evaluation. The results of the fusion performance of the proposed method and the other three methods are shown in Figure 7. Figure 7(a) is the visible image, Figure 7(b) is the infrared image, and Figures 7(c)–7(f) show the fusion results of DenseFuse [28], DeepFuse [29], FusionGAN [16], and the proposed method, respectively. Intuitively, all four methods fuse the texture information of the visible image and the thermal radiation information of the infrared image together to some extent. However, the fusion results of our method are more closely aligned with human visual perception, better preserve visible information, and retain more infrared information, making the image look richer and clearer with higher contrast. In addition, the target area is also more prominent than those of the other three methods.

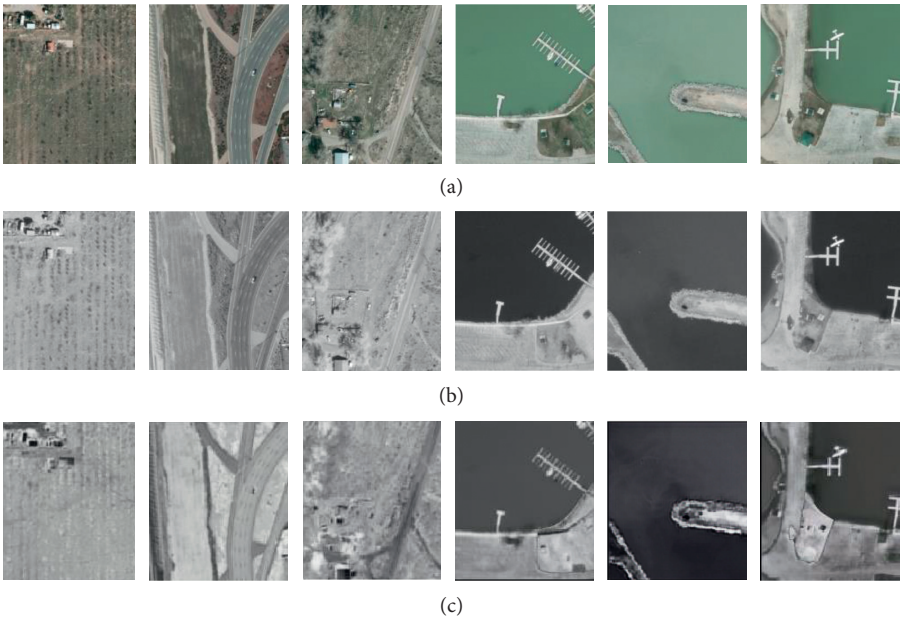


Figure 9: Infrared images generated using the proposed method.

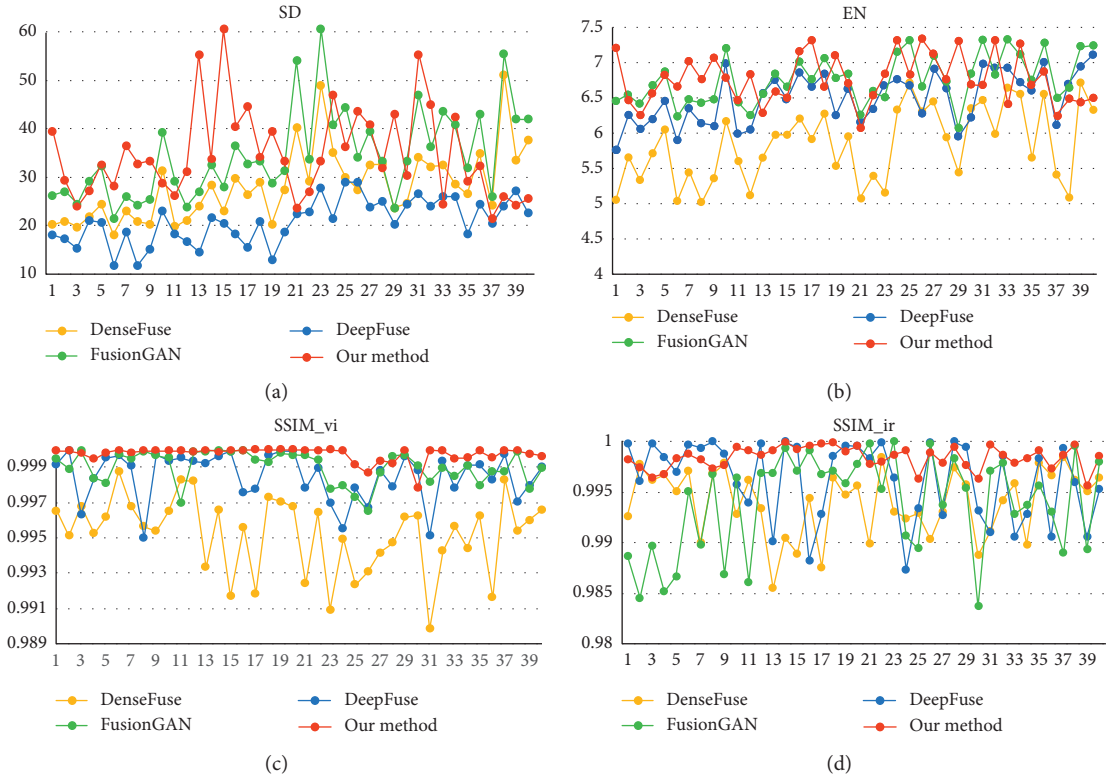


Figure 10: Continued.

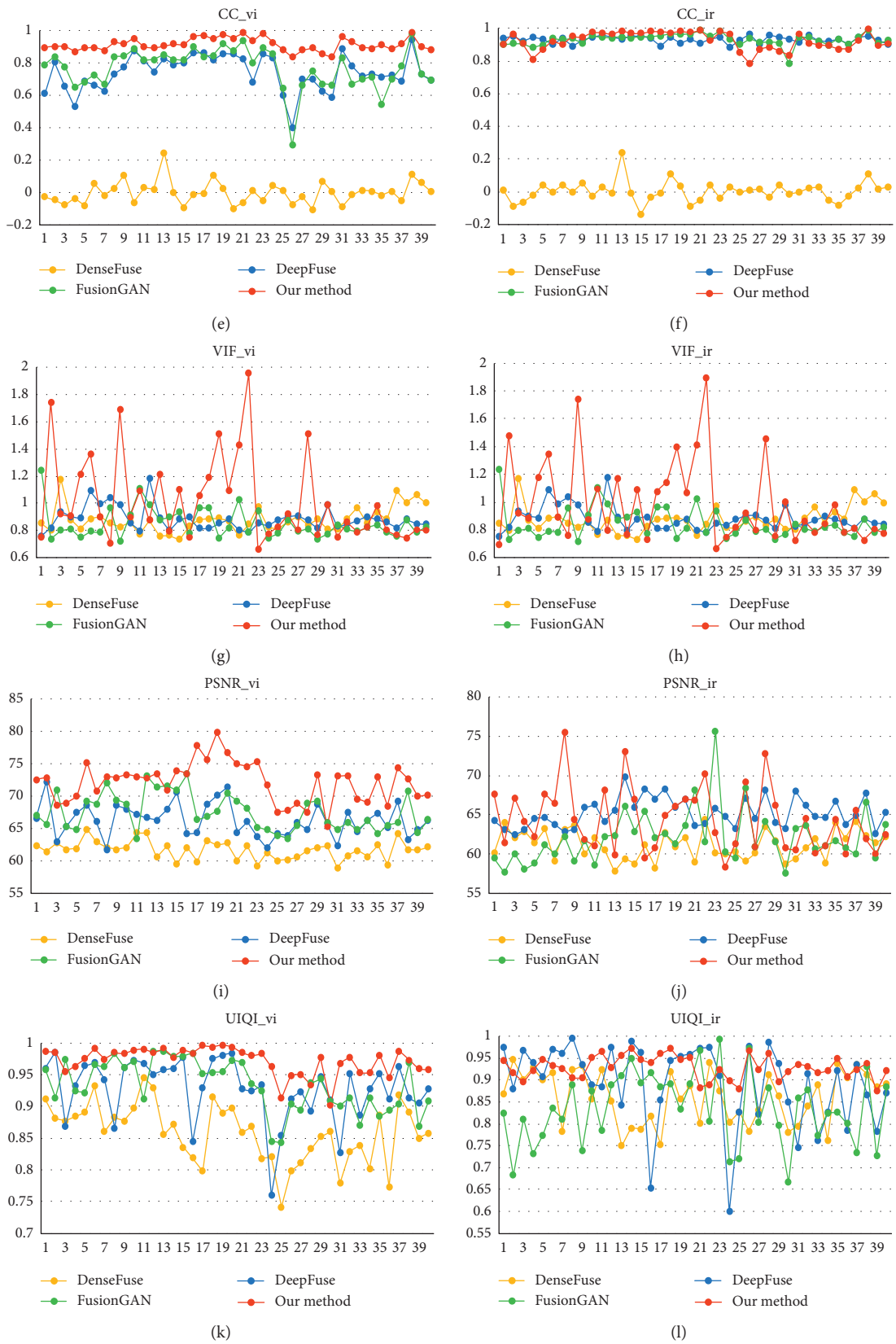


Figure 10: Continued.

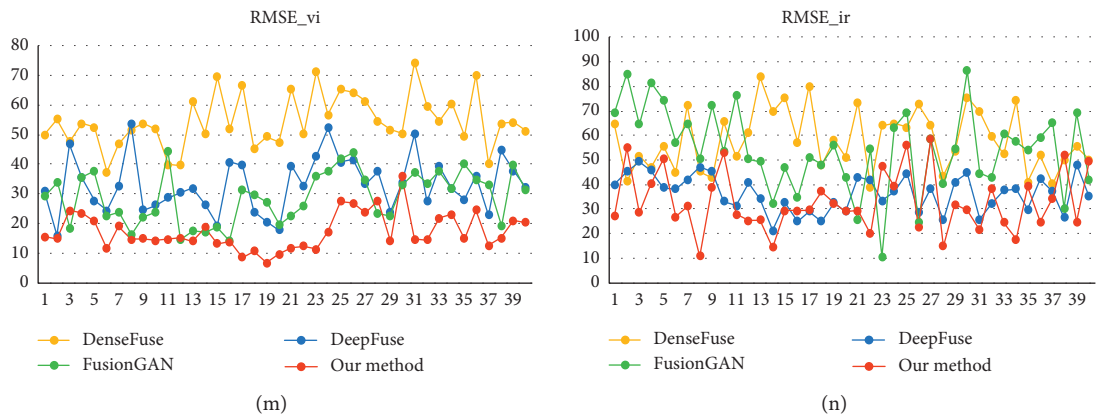


Figure 10: Quantitative comparison of the 14 dimensions of 8 indexes for the images from the VEDAI dataset.

Quantitative comparison: the qualitative illustrations in Figure 7 cannot objectively determine the quality of the results. Therefore, the fusion methods were further compared using quantitative methods. Eight indexes were used on 56 pairs of images from the TNO dataset, of which six indexes require a reference (the fused image refers to the corresponding visible and infrared images). The results are shown in Figure 8. The proposed method achieves the best performance for the majority of image pairs, and, for some individual image pairs, the comprehensive fusion index is much higher than that of the other methods. In addition, compared with the other three methods, the proposed method has the best average of the evaluation indexes. Because the proposed method uses two discriminants when referring to visible images, it is comparable to the FusionGAN method. However, when referring to infrared images, the proposed method considerably outperforms the FusionGAN method. This shows that the proposed method retains more infrared thermal radiation information while retaining sufficient visible texture information. Thus, our training framework is both effective and essential.

4.2.2. Validation by the VEDAI Dataset. The Vehicle Detection in Aerial Imagery (VEDAI) dataset [30] contains publicly available aerial orthogonal normalization images from Utah's State Geographic Information Database (SGID) by the Automated Geographic Reference Center (AGRC). These aerial orthogonal normalization images contain a wide variety of vehicles, backgrounds, and obfuscated objects. Each image has three visible channels and one near-infrared channel. In this section, the DenseFuse, DeepFuse, FusionGAN, and proposed method are further tested using the VEDAI dataset. Figure 9 shows the proposed method's generation of infrared images from visible images. Figure 9(a) shows the visible images, Figure 9(b) shows the actual infrared images corresponding to the visible images from the dataset, and Figure 9(c) shows the infrared images generated directly from the visible images using the proposed method. Figure 9 illustrates that the infrared images generated by the proposed method accurately reflect actual thermal radiation information while maintaining consistency with the real infrared images.

A total of 40 images from the VEDAI dataset were selected for a quantitative comparison. Figure 10 shows a quantitative analysis of the fusion results of the four methods using 8 evaluation indexes. The proposed method achieves the best SSIM, CC, PSNR, UIQI, and RMSE results on the majority of images. Compared with the other three methods, the averages of the other indexes of the proposed method are the highest. Our experiments show that the proposed method generalizes well to other datasets.

5. Conclusion

In this paper, we propose a new fusion method that generates a matched infrared image from a visible image and generates a fused image that retains more visible texture details and infrared heat radiation information than other methods. Experimental evaluations on two public datasets show that the proposed method generates infrared images with thermal radiation information relatively consistent with real infrared images and generates fused images with clearly prominent texture information and rich thermal radiation information. A quantitative analysis of eight evaluation indexes for fused images shows that the proposed method produces better visual effects while retaining more information than other methods.

Data Availability

All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] T. Xiang, L. Yan, and R. Gao, "A fusion algorithm for infrared and visible images based on adaptive dual-channel unit-linking PCNN in NSCT domain," *Infrared Physics & Technology*, vol. 69, pp. 53–61, 2015.

[2] W. Kong, L. Zhang, and Y. Lei, "Novel fusion method for visible light and infrared images based on NSST-SF-PCNN," *Infrared Physics & Technology*, vol. 65, pp. 103–112, 2014.

[3] S. Li, B. Yang, and J. Hu, "Performance comparison of different multi-resolution transforms for image fusion," *Information Fusion*, vol. 12, no. 2, pp. 74–84, 2011.

[4] G. Pajares and J. Manuel de la Cruz, "A wavelet-based image fusion tutorial," *Pattern Recognition*, vol. 37, no. 9, pp. 1855–1872, 2004.

[5] Z. Zhang and R. S. Blum, "A categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application," *Proceedings of the IEEE*, vol. 87, no. 8, pp. 1315–1326, 1999.

[6] D. P. Bavirisetti, G. Xiao, and G. Liu, "Multi-sensor image fusion based on fourth order partial differential equations," in *Proceedings of the 20th International Conference on Information Fusion (Fusion)*, pp. 1–9, Xi'an, China, October 2017.

[7] W. Kong, Y. Lei, and H. Zhao, "Adaptive fusion method of visible light and infrared images based on non-subsampled shearlet transform and fast non-negative matrix factorization," *Infrared Physics & Technology*, vol. 67, pp. 161–172, 2014.

[8] X. Zhang, Y. Ma, F. Fan, Y. Zhang, and J. Huang, "Infrared and visible image fusion via saliency analysis and local edge-preserving multi-scale decomposition," *Journal of the Optical Society of America A*, vol. 34, no. 8, pp. 1400–1410, 2017.

[9] J. Zhao, Y. Chen, H. Feng, Z. Xu, and Q. Li, "Infrared image enhancement through saliency feature analysis based on multi-scale decomposition," *Infrared Physics & Technology*, vol. 62, pp. 86–93, 2014.

[10] J. Wang, J. Peng, X. Feng, G. He, and J. Fan, "Fusion method for infrared and visible images by using non-negative sparse representation," *Infrared Physics & Technology*, vol. 67, pp. 477–489, 2014.

[11] S. Li, H. Yin, and L. Fang, "Group-sparse representation with dictionary learning for medical image denoising and fusion," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 12, pp. 3450–3459, 2012.

[12] J. Zhao, G. Cui, X. Gong, Y. Zang, S. Tao, and D. Wang, "Fusion of visible and infrared images using global entropy and gradient constrained regularization," *Infrared Physics & Technology*, vol. 81, pp. 201–209, 2017.

[13] J. Ma, Z. Zhou, B. Wang, and H. Zong, "Infrared and visible image fusion based on visual saliency map and weighted least square optimization," *Infrared Physics & Technology*, vol. 82, pp. 8–17, 2017.

[14] J. Ma, C. Chen, C. Li, and J. Huang, "Infrared and visible image fusion via gradient transfer and total variation minimization," *Information Fusion*, vol. 31, pp. 100–109, 2016.

[15] Y. Liu, S. Liu, and Z. Wang, "A general framework for image fusion based on multi-scale transform and sparse representation," *Information Fusion*, vol. 24, pp. 147–164, 2015.

[16] J. Ma, W. Yu, P. Liang, C. Li, and J. Jiang, "FusionGAN: a generative adversarial network for infrared and visible image fusion," *Information Fusion*, vol. 48, pp. 11–26, 2019.

[17] I. Goodfellow, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, Morgan Kaufmann Publishers, Burlington, MA, USA, 2014.

[18] H. Li, X.-J. Wu, and J. Kittler, "Infrared and visible image fusion using a deep learning framework," in *Proceedings of the 24th International Conference on Pattern Recognition (ICPR)*, IEEE, Beijing, China, pp. 2705–2710, 2018.

[19] Y. Zhang, L. Zhang, X. Bai, and L. Zhang, "Infrared and visual image fusion through infrared feature extraction and visual information preservation," *Infrared Physics & Technology*, vol. 83, pp. 227–237, 2017.

[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, Honolulu, HI, USA, July 2017.

[21] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, https://arxiv.org/abs/1411.1784.

[22] Y.-J. Rao, "In-fibre Bragg grating sensors," *Measurement Science and Technology*, vol. 8, no. 4, pp. 355–375, 1997.

[23] J. W. Roberts, J. A. Van Aardt, and F. B. Ahmed, "Assessment of image fusion procedures using entropy, image quality, and multispectral classification," *Journal of Applied Remote Sensing*, vol. 2, no. 1, Article ID 023522, 2008.

[24] M. Deshmukh and U. Bhosale, "Image fusion and image quality assessment of fused images," *International Journal of Image Processing (IJIP)*, vol. 4, no. 5, p. 484, 2010.

[25] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proceedings of the International Conference on Pattern Recognition*, IEEE, Istanbul, Turkey, pp. 2366–2369, August 2010.

[26] Y. Han, Y. Cai, Y. Cao, and X. Xu, "A new image fusion performance metric based on visual information fidelity," *Information Fusion*, vol. 14, no. 2, pp. 127–135, 2013.

[27] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, 2002.

[28] H. Li and X.-J. Wu, "DenseFuse: a fusion approach to infrared and visible images," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2614–2623, 2018.

[29] K. R. Prabhakar, V. S. Srikar, and R. V. Babu, "DeepFuse: a deep unsupervised approach for exposure fusion with extreme exposure image pairs," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4724–4732, Venice, Italy, December 2017.

[30] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery : a small target detection benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2016.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 4726763, 18 pages
https://doi.org/10.1155/2020/4726763

Research Article

The Driver Time Memory Car-Following Model Simulating in Apollo Platform with GRU and Real Road Traffic Data

Rong Fei,[1] **Shasha Li**,[1] **Xinhong Hei**,[1] **Qingzheng Xu**,[2] **Fang Liu**,[3] **and Bo Hu**[4]

[1]*School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China*
[2]*College of Information and Communication, National University of Defense Technology, Xi'an 710106, China*
[3]*School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China*
[4]*Beijing Huadian Youkong Technology Co., Ltd., Beijing 100193, China*

Correspondence should be addressed to Rong Fei; annyfei@xaut.edu.cn and Xinhong Hei; heixinhong@xaut.edu.cn

Received 15 January 2020; Accepted 11 February 2020; Published 17 March 2020

Guest Editor: Chi-Hua Chen

Car following is the most common phenomenon in single-lane traffic. The accuracy of acceleration prediction can be effectively improved by the driver's memory in car-following behaviour. In addition, the Apollo autonomous driving platform launched by Baidu Inc. provides fast test vehicle following vehicle models. Therefore, this paper proposes a car-following model (CFDT) with driver time memory based on real-world traffic data. The CFDT model is firstly constructed by embedded gantry control unit storage capacity (GRU assisted) network. Secondly, the NGSIM dataset will be used to obtain the tracking data of small vehicles with similar driving behaviours from the common real road vehicle driving tracks for data preprocessing according to the response time of drivers. Then, the model is calibrated to obtain the driver's driving memory and the optimal parameters of the model and structure. Finally, the Apollo simulation platform with high-speed automatic driving technology is used for 3D visualization interface verification. Comparative experiments on vehicle tracking characteristics show that the CFDT model is effective and robust, which improves the simulation accuracy. Meanwhile, the model is tested and validated using the Apollo simulation platform to ensure accuracy and utility of the model.

1. Introduction

Car-following (CF) behaviour is the most basic micro driving behaviour, referring to the interaction between two adjacent vehicles in a vehicle fleet driving on a single-lane road that does not allow passing [1]. The concept of CF originated in the early 1950s. Over the past six decades, CF models have been extensively and systematically studied, and fruitful achievements have been made [2]. Since the 1990s, research in related fields has gradually emerged in China. Researchers from various fields have attempted to interpret the observed microscopic phenomena from different perspectives [3].

There are currently many types of CF models that can be divided into two categories based on their origins: theory-driven and data-driven CF models [4].

In the development of the theory-driven models, the stimulus-response models are the most classic CF models, of which the General Motors (GM) model [5] is the most important. The GM model has been gradually developed and used since the late 1950s; it is the basis of many of the subsequent stimulus-response models. The GM model clearly reflects the characteristics of CF behaviour; it has a simple form and a clear physical meaning based on its originality. However, this model is prone to change with changes in traffic operational conditions and hence lacks universality.

With the increasing popularity of artificial intelligence, data-driven models have gradually become a focus of research of CF models. In 1988, Rumelhart proposed the back-propagation neural network (BPNN) [6], which is a multilayer feedforward neural network (FNN) that uses the error back-propagation algorithm to adjust weights. It is the most widely used NN model. Chen et al. proposed a deep learning method for learning potentially complex and irregular

probability distributions, which can accurately estimate the values of CDF and PDF [7].

With the wide application of NNs in the field of traffic simulation, in 1998, Kehtarnavaz [8] applied the BPNN model to CF behaviour modelling for the first time, using the speed of the following car and the distance between the two cars as the model inputs and the relative speed of the two cars as the model output, thus verifying the validity of the model. Zhang et al. [9] established a closed-loop driving following model based on BP neural network and verified the adaptability of the model to different driving groups through experiments.

Support vector regression (SVR) [10] is a machine learning algorithm that converts the original problem into a convex quadratic programming problem and solves it using the optimality theory to obtain the global optimal solution. Wei and Liu [11] proposed a vehicle following model based on support vector regression and studied the asymmetric characteristics of vehicle following behaviour and its influence on the evolution of traffic flow. Parham et al. [12] propose car-following modelling using an efficient support vector regression method and prove that it has appropriate validity after inputting the driver's instantaneous reaction time. However, studies based on such models are in their infancy.

CF behaviour is a continuous behaviour, so a driver can make a corresponding decision based on the memory of the previous time period [13–15]. However, a large number of existing models do not fully consider the driver's memory effect and only consider the instantaneous interaction between the following and leading cars. To process the CF time series data to use their historical information, the model must have a memory capability; this is lacking in both the BPNN- and SVR-based models.

Recurrent neural network (RNN) [16] is a class of neural network with memory capability. Yang [17] proposed a car-following model based on recurrent neural network (RNN) to effectively describe the state changes of vehicles while driving and road traffic congestion.

When an input sequence is long, the gradient explosion and vanishing problems, also known as the long-term dependency problem, will occur. To solve this problem, various modifications have been made on RNNs; the most effective method is to introduce various gating mechanisms such as the long short-term memory (LSTM) [18, 19] and the gated recurrent unit (GRU) [20] networks.

Based on the previously described studies, Wang et al. [21] proposed the use of GRU to model CF behaviour and embed the driver's memory effect in the model, which used the speed of the following car, the relative speed of the two cars, and the distance between the two cars observed in the last several time intervals as inputs and the estimated speed of the following car at the next time point as the output. The test results showed that the proposed model has higher simulation accuracy than the existing CF models and provides a new concept for the study of traffic flow theory and simulation. However, the driver's decision-making and reflection time are not considered in the judgment process.

However, influenced by multiple sources of information, a driver's decision-making and judgment process exhibits a complex nonlinear modality during driving, and the driver's psychological decision cannot be described with a simple mathematical expression. Fuzzy theories and artificial neural networks show certain operational advantages in handling complex nonlinear issues and also exhibit a good learning capacity under big data samples. Therefore, the fuzzy theory and artificial neural network are often used for simulating driving behaviours under different environments. However, the current schemes utilizing fuzzy theories and artificial neural networks only focus on the velocities and accelerations of the leading car and the following car, as well as the spacing there between, without considering driving environments [22]. In addition, how to obtain real-time traffic information (such as average speed, travel time, traffic flow, and traffic conditions) is also an important issue for unmanned driving. Many scholars have also done a lot of research on real-time traffic information. Chen proposed a cell probe (CP)-based method to analyse cellular network signals with an estimated accuracy of 97.63%, which is easier to obtain than traditional methods [23].

In April 2017, Baidu released its open platform Apollo for autonomous driving; after iterations of multiple versions, the platform has been enabled for localization, sensing, decision, and simulation. Apollo may help its partners in the automotive and autonomous driving industries to quickly develop a set of their own autonomous driving systems in consideration of vehicles and hardware systems. In the Apollo simulation environment, environment information including traffic signs, index lines, and the relationships with surrounding vehicles may be inputted into Dreamview via corresponding interfaces to thereby construct a driving environment. Besides, the Apollo platform is further enabled for validating the car-following model and optimizing the relevant algorithm through a 3D visual interface.

In this study, based on the previously described studies and combined with actual road conditions, we designed a CFDT model based on the data-driven model in combination with the improved RNN. In our model, the speed of the leading vehicle in the previous time interval, the speed of the following vehicle, and the distance between the two vehicles are used as inputs to predict the acceleration of the following vehicle at the next time point. Furthermore, the established model was calibrated using the CF data to determine the optimal parameters and optimal structure of the model, which were then verified through simulation. Finally, the proposed model was compared with the BPNN- and SVR-based models. It was confirmed that, compared with the traditional CF models, the RNN network-based CF model has high robustness and improved simulation accuracy, providing a methodological basis for studying the car-following behaviour.

The remainder of the paper is divided into four sections. Section 2 introduces conventional car-following models and the neural network based car-following model. Section 3 models CF behaviour mainly using the RNN network. Section 4 processes the data and briefly analyses the driver's response time. Section 5 trains the proposed model to obtain the optimal parameters, compares it with the other two

existing models, and verifies that the proposed model can achieve better simulation results. Section 6 conducts an empirical study of the three types of CF models based on the data and describes in detail the models' verification experiments. Section 7 uses the Apollo simulation platform to verify the model and ensure the accuracy and practicability of the model. Section 8 introduces the summary and prospects.

2. Background

2.1. Traditional Car-Following Models. The stimulus-response framework is the most traditional modelling idea of the car-following behaviour, which embodies many essential characteristics of the car-following behaviour, while the GM model is the most important stimulus-response type of the traditional vehicle following model. The GM model assumes that a vehicle does not show passing or lane-changing behaviour when following. The driving dynamics theory is used to derive the basic equation:

$$a_{n+1}(t+T) = \lambda v_{n+1}^{m}(t+T)\frac{\Delta v(t)}{[\Delta x(t)]^{l}}. \quad (1)$$

In equation (1), $a_{n+1}(t+T)$ is the instantaneous acceleration of a following car at time $(t+T)$; $v_{n+1}(t+T)$ is the instantaneous speed of a following car at time $(t+T)$; $\Delta v(t)$ is the relative speed of the two cars at time t; $\Delta x(t)$ is the distance between the two cars at time t; T is the response time; λ is the sensitivity parameter to be calibrated; and m and l are additional parameters to be calibrated. Numerous studies have been focused on the parameter calibration and extension of the GM model.

The GM model clearly reflects the characteristics of CF behaviour; it has a simple form and a clear physical meaning based on its originality. However, this model is prone to change with changes in traffic operational conditions and hence lacks universality.

2.2. Basics of NN. NN is a highly nonlinear model with a neuron as its basic unit. When a neuron receives a set of input signals, it generates an output signal. A typical structure is shown in Figure 1.

where $x = [x_1 \ x_2 \ \cdots \ x_d]^T \in \mathbb{R}^d$ is the input, $w = [w_1 \ w_2 \ \cdots \ w_d]^T \in \mathbb{R}^d$ is the weight, and b is a bias unit. The sum of the weighted inputs is described using the net input z; then, we have equation (2), as follows:

$$z = \mathbf{w}^T\mathbf{x} + b, \quad (2)$$

where $f(x)$ is the activation function. Then, the output can be expressed as follows in equation (3):

$$a = f(z). \quad (3)$$

The commonly used activation functions, which are nonlinear, are as follows:

Sigmoid function:

$$\sigma(x) = \frac{1}{1+e^{-x}}. \quad (4)$$

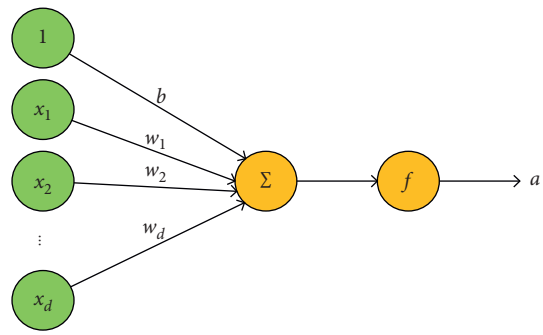


Figure 1: Structure of a neuron.

Tanh function:

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}. \quad (5)$$

ReLU function:

$$\mathrm{ReLU}(x) = \max(0, x). \quad (6)$$

Leaky ReLU function:

$$\mathrm{LeakyReLU}(x) = \max(\alpha x, x). \quad (7)$$

In practical application, the appropriate activation function can be selected according to the actual situation.

Historically, various NN structures have been proposed; those most commonly used include the feedforward NN, the feedback NN, and the graph network. In this study, the feedback NN was adopted. The neurons in a feedback NN can receive not only the signals of other neurons but also their own. Compared with those in a feedforward NN, the neurons in a feedback NN have a memory function and have different states at different times. The basic structure of a feedback NN is shown in Figure 2.

In a feedback NN, signals can propagate in one or both directions. This type of network includes RNN and the Boltzmann machine.

2.3. Gated RNN. To solve the long-term dependence problem of RNNs in the long sequence of training, a gating mechanism is introduced to selectively add new information while selectively forgetting the retained information. Such networks are collectively referred to as gated RNNs; the most popular include the LSTM and GRU networks.

2.3.1. LSTM Network. An LSTM network adds the new internal states $\mathbf{c}_t$ and introduces three "gates"—the forgetting gate ($\mathbf{f}_t$), input gate ($\mathbf{i}_t$), and output gate ($\mathbf{o}_t$). The value of a "gate" is within $(0, 1)$; it is used to control the amount of information passed.

Specifically, the forgetting gate $\mathbf{f}_t$ controls the amount of information to be forgotten by the internal state of the last time point ($\mathbf{c}_{t-1}$); the input gate $\mathbf{i}_t$ controls the amount of information to be retained by the candidate state $\tilde{c}_t$ of the current time point; and the output gate $\mathbf{o}_t$ controls the

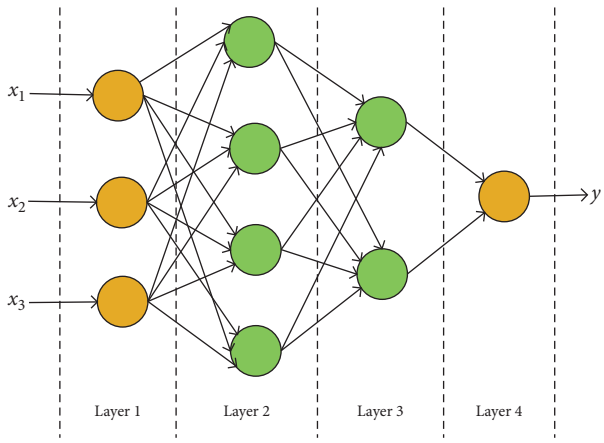


Figure 2: Feedback NN.

amount of information to be output to the external state $\mathbf{h}_t$ by the internal state $\mathbf{c}_t$ of the current time point.

Figure 3 shows the internal structure of the LSTM unit, where $\times$ represents the multiplication of vector elements and $+$ represents the addition of vector elements; $\sigma(x)$ is the sigmoid activation function. Thus, the three gates of $\mathbf{f}_t$, $\mathbf{i}_t$, and $\mathbf{o}_t$ can be calculated with equations (8)–(10), respectively. $W_*(* = f, i, o)$ and $U_*(* = f, i, o)$ stand for the weight matrix from the cell to gate, and $b_*(* = f, i, o)$ denotes the vector of each gate.

$$f_t = \sigma\left(U_f h_{t-1} + W_f x_t + b_f\right), \tag{8}$$

$$i_t = \sigma\left(U_i h_{t-1} + W_i x_t + b_i\right), \tag{9}$$

$$o_t = \sigma\left(U_o h_{t-1} + W_o x_t + b_o\right). \tag{10}$$

The methods for status updating are described in equations (11)–(13) for $\mathbf{f}_t$, $\mathbf{i}_t$, and $\mathbf{o}_t$, respectively:

$$\tilde{c}_t = \tanh\left(U_c h_{t-1} + W_c x_t + b_c\right), \tag{11}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t, \tag{12}$$

$$h_t = o_t * \tanh\left(c_t\right). \tag{13}$$

In equations (12) and (13), *represents the multiplication of vector elements.

In an LSTM network, the internal state of the unit $\mathbf{c}_t$ can retain certain key information for a significant amount of time.

2.3.2. GRU Network. The internal concept of a GRU network is similar to that of an LSTM network, and a GRU network can achieve a comparable effect. However, a GRU network has fewer parameters, lower training difficulty, and higher practicality.

Because the input and forgetting gates in the LSTM unit are complementary, in a GRU unit, they are combined into one gate, i.e., the update gate $\mathbf{z}_t$, while the output gate of the LSTM unit is deleted and a reset gate $\mathbf{r}_t$ is added without introducing a new internal state. Its structure is shown in Figure 4.

Where the update gate controls the amount of information to be retained by the state of the current time point $(\mathbf{h}_t)$ from the state of the last time point $(\mathbf{h}_{t-1})$, as well as the amount of new information received from the candidate state $(\tilde{h}_t)$, the reset gate controls the amount of information to be retained from the state of the last time point $(\mathbf{h}_{t-1})$ by the candidate state of the current time point $(\tilde{h}_t)$. The methods for the calculation of the two gates are shown in equations (14) and (15). $W_*(* = z, r, h)$ and $U_*(* = z, r, h)$ stand for the weight matrix from the cell to gate, and $b_*(* = z, r, h)$ denotes the vector of each gate.

$$z_t = \sigma\left(U_z h_{t-1} + W_z x_t + b_z\right), \tag{14}$$

$$r_t = \sigma\left(U_r h_{t-1} + W_r x_t + b_r\right). \tag{15}$$

The methods for updating the states are shown in equations (16) and (17):

$$\tilde{h}_t = \tanh\left(U_h\left(r_t * h_{t-1}\right) + W_h x_t + b_h\right), \tag{16}$$

$$h_t = z_t * h_{t-1} + \left(1 - z_t\right) * \tilde{h}_t. \tag{17}$$

In the case where $\mathbf{z}_t = 0$ and $\mathbf{r}_t = 1$, the GRU network degenerates into a simple RNN.

3. Approach to the Proposed Model

Herein, we adopted the RNN network to model CF behaviour. The proper choice of model inputs and outputs can improve the simulation accuracy of the model. Based on the GM model, we use the speed of the leading car at time t $(v_n(t))$, the speed of the following car at time t $(v_{n+1}(t))$, and the distance between the two cars $(\Delta x(t))$ as inputs and the acceleration of the following car at time $t + T$ $(a_{n+1}(t + T))$ as the output. Then, we have equation (18):

$$a_{n+1}(t+T) = f\begin{pmatrix} v_n(t), v_{n+1}(t), \Delta x(t), v_n(t-T), v_{n+1}(t-T), \Delta x(t-T), \ldots \\ v_n(t-(N-1)T),\ v_{n+1}(t-(N-1)T), \Delta x(t-(N-1)T) \end{pmatrix}, \tag{18}$$

where N is the length of the time interval of "memory." Figure 5 is the specific structure diagram of the model proposed in this paper.

To eliminate the influence of dimension on the simulation accuracy and convergence rate of the model, the paper normalizes the vehicle following data (leading vehicle speed, following vehicle speed, and following vehicle acceleration). In different traffic environments, the behaviour of following a car is easily affected by the propagation of the slight disturbance of the speed of the leading car, the habit of acceleration and deceleration, the driver's cognition of the environment, and the driver's cognition of driving

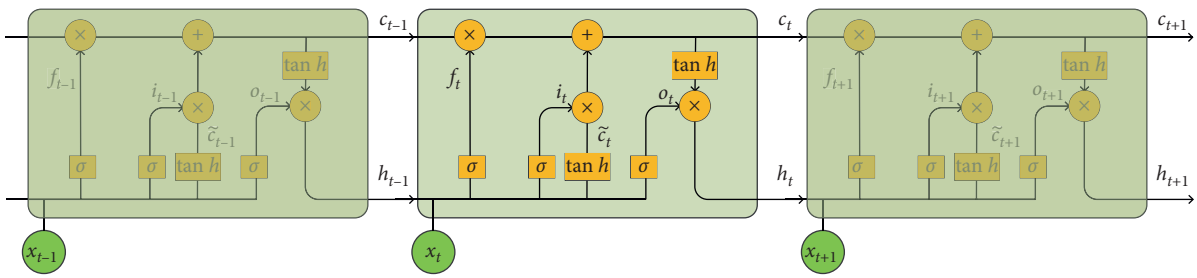


Figure 3: LSTM unit.

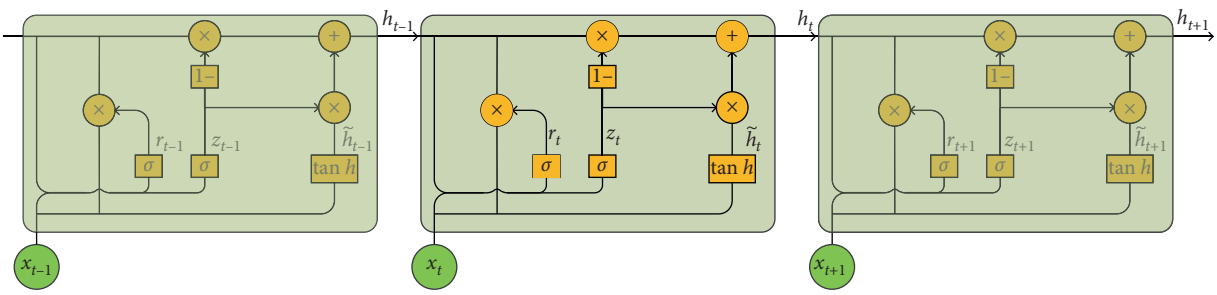


Figure 4: GRU unit.

behaviour. The distribution of the car-following data cannot judge whether it is close to a Gaussian distribution. Therefore, the MinMaxScaler of the simplest method to eliminate the influence of dimensionality and data value range is selected in this paper to preserve the relationship existing in the original data. The formula for MinMaxScaler is as follows:

$$X_{\text{scaled}} = \frac{X - X.\text{min}}{X.\text{max} - X.\text{min}} \cdot (\text{max} - \text{min}) + \text{min}, \tag{19}$$

where max and min are the maximum and minimum values of a given zoom range, respectively; i.e., the original data are scaled to the range of [max, min]. Specifically, the original data are scaled to the range of [0, 1]. Then, we have the following:

$$X_{\text{scaled}} = \frac{X - X.\text{min}}{X.\text{max} - X.\text{min}}. \tag{20}$$

The CF data are constructed to the appropriate input and output shapes to comply with the input and output structures of the GRU model. Because the length of the "memory" interval (N) is closely related to the constructed data, it also directly affects the prediction accuracy of the model and so it was tested in detail in this study. Then, the dataset was randomly divided into training and test sets. Next, ReLU was selected as the activation function of the output layer to establish the GRU model. Because the number of hidden layers and the number of neurons in each hidden layer can have numerous combinations, it is necessary to separately test each of the models with different structures to obtain the optimal structure of the model.

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2. \tag{21}$$

The mean square error (MSE) (21) was used to construct the loss function and the Adam optimizer [24–26], due to its excellent performance in most cases, was adopted.

In the input layer, the time step is set to N, and each step contains three input variables $(v_n, v_{n+1}, \Delta x)$. Since it is necessary to predict the acceleration of the following vehicle with a continuous time of m, the input value is constructed into a matrix $X \in \mathbb{R}^{m \times N \times 3}$ through the normalization of formula (19). Finally, the predicted acceleration of d for a continuous period of time of the following car was constructed into output $y \in \mathbb{R}^{m \times 1}$ through the GRU gate unit of multiple hidden layers. The pseudocode for the construction of the RNN-based CF model is shown in Algorithm 1.

4. Data Preparation

4.1. Processing of the Following Vehicle Data. The Next Generation Simulation (NGSIM) program [27] was initiated by the U.S. Federal Highway Administration (FHWA). Through the established synchronous digital camera network, detailed vehicle trajectory data were acquired at a time interval of 0.1 seconds from the US-101 Freeway and the southbound direction of Lankershim Boulevard in Los Angeles, California; Interstate I-80 in Emeryville, California; and the eastbound direction of Peachtree Street in Atlanta, Georgia.

In this study, the detailed trajectory data of the eastbound vehicles on Interstate I-80 in Emeryville, California, were used. The data were acquired at 10 frames per second by seven cameras mounted on the 30-story Pacific Park Plaza Building located at Christie Avenue. The study road section is 503 m long and has six lanes. Lane 1 is a high occupancy vehicle (HOV) lane and Lane 6 is a collector-distributor lane, as shown in Figure 6.

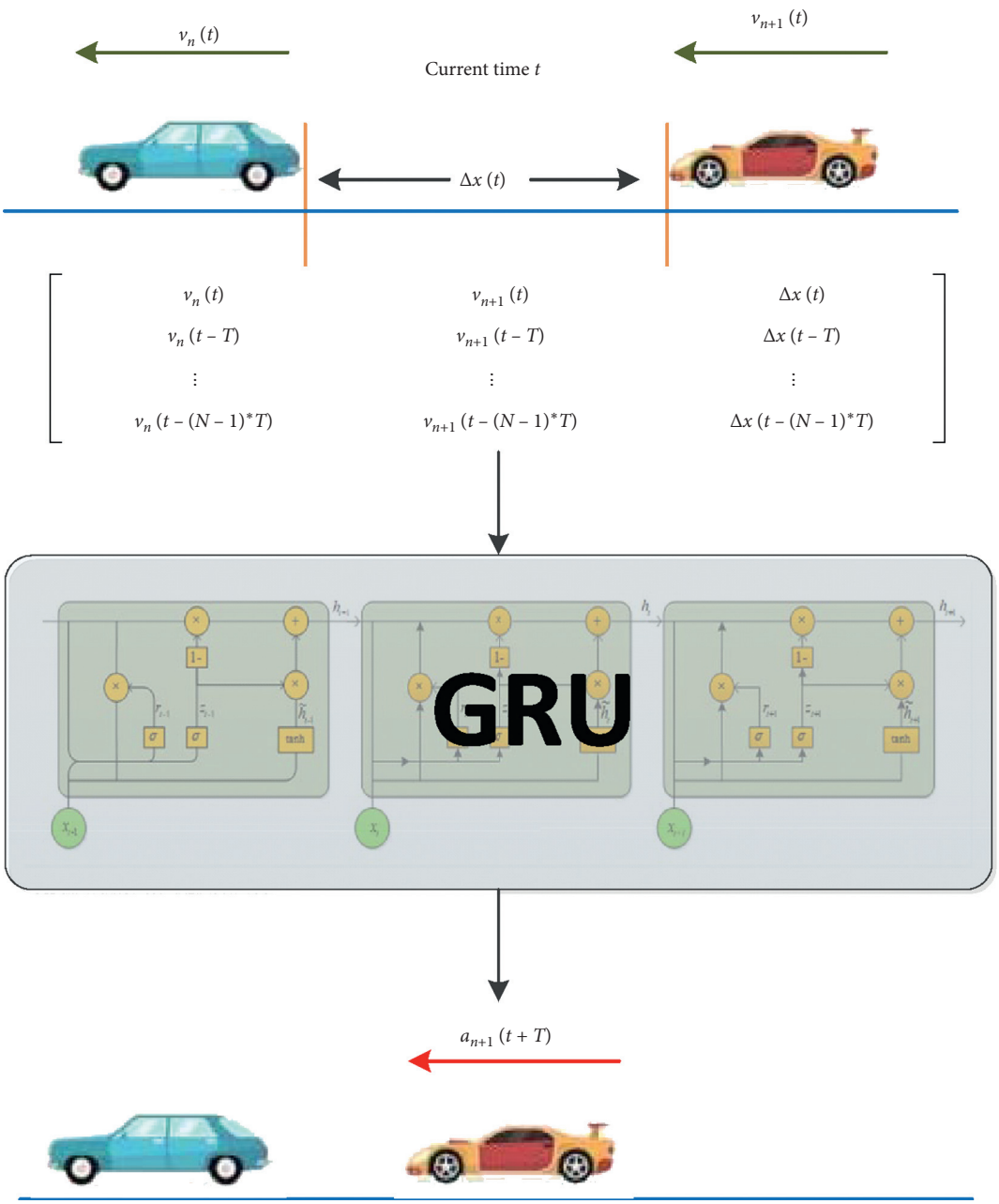


Figure 5: Schematic structure of the GRU-based CF model.

The I-80 dataset includes vehicle trajectory data for three time periods. The road conditions are shown in Table 1.

In this study, we mainly analysed the CF behaviour of cars and their microscopic characteristics. The CF behaviour is closely related to the road conditions. To ensure the universality of the CF behaviour, we examined the vehicle trajectory data between 4:00 pm and 4:15 pm, which contain 1,028,575 trajectory records. Each record contains 18 fields, as fully described in Table 2.

4.1.1. Data Preprocessing. First, data of the following vehicles were found and filtered according to the following rules:

(1) To avoid the potential difference in CF behaviour between different types of vehicles, we only focused on the CF behaviour of small cars (i.e., v_Class = 2)

(2) Cars from the HOV lane (i.e., Lane_ID = 1) and the collector-distributor lane (i.e., Lane_ID = 6) were excluded to ensure that the vehicles under study are associated with similar driving behaviour, i.e., to ensure the consistency of driving behaviour

(3) The single-lane data of driving vehicles were adopted to avoid the influence of lane-changing behaviour on CF behaviour

(4) Only the data for cars with a following time greater than 45 s (i.e., 450 records) were retained to ensure

GRU-based car-following model
Input: $v_n(t), v_{n+1}(t), \Delta x(t), v_n(t-T), v_{n+1}(t-T), \Delta x(t-T), \ldots,$
$v_n(t-(N-1)T),\ v_{n+1}(t-(N-1)T), \Delta x(t-(N-1)T)$
Output: $a_{n+1}(t+T)$

```
Data normalization using MinMaxScaler.
The input is constructed as X ∈ ℝ^{m×N×3}, The output is constructed as y ∈ ℝ^{m×1}
The constructed data is divided into training set and testing set.
Modelling Sequential(), add GRU Layer and Dense Layer
Compiler model, loss function = "MSE," optimizer = "Adam"
while model convergence do
        Training model
end while
Test model
```

Algorithm 1: Pseudocode for the construction of the GRU-based CF model.

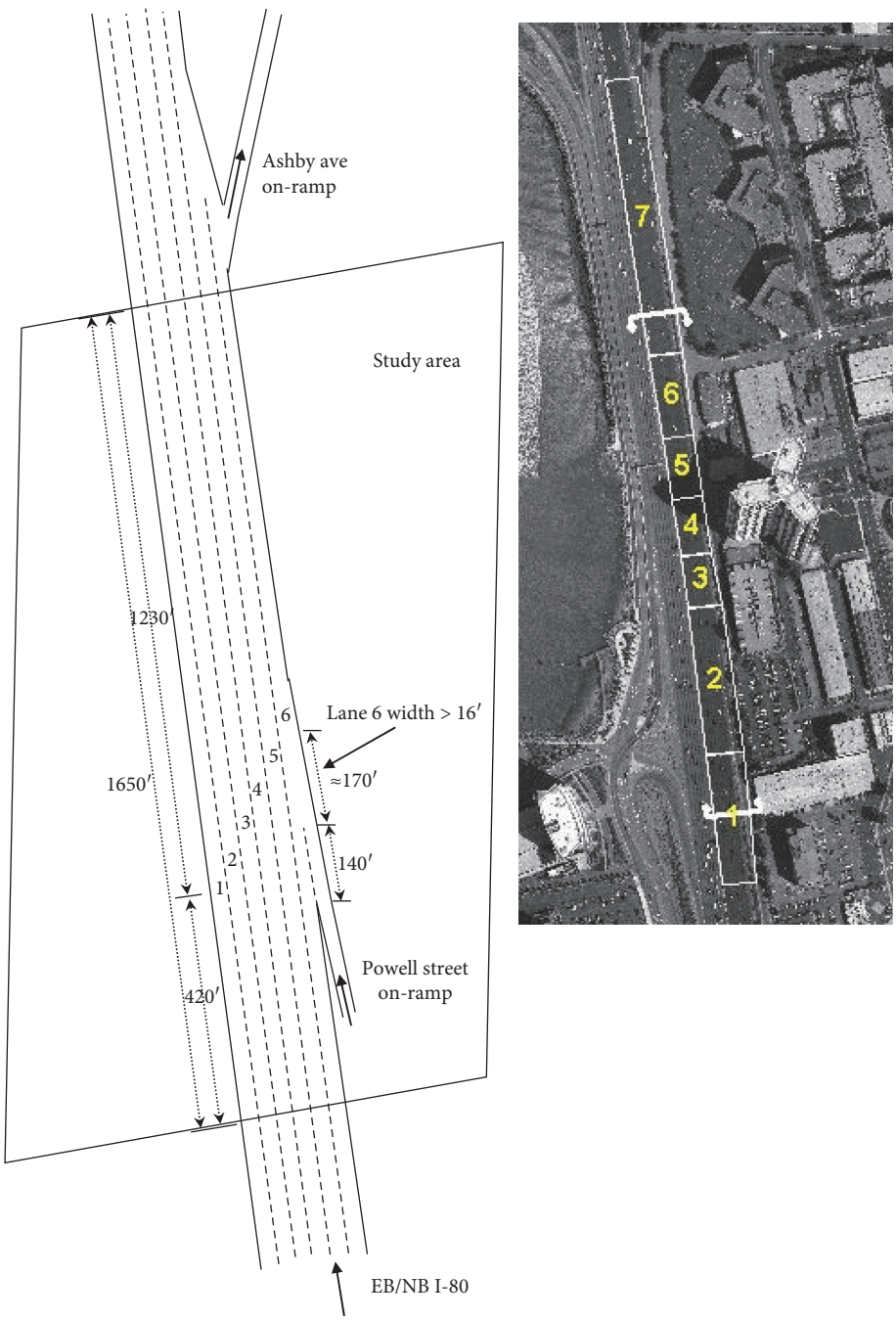


Figure 6: Outline of the study area.

Table 1: Road conditions.

Acquisition period	Road condition
4:00 p.m.~4:15 p.m.	Transition from noncongestion to congestion
5:00 p.m.~5:15 p.m.	Congestion
5:15 p.m.~5:30 p.m.	Congestion

Table 2: Field descriptions.

Field	Description
Vehicle_ID	Vehicle identification number (increase by entry time)
Frame_ID	Frame identification number (increase by start time)
Total_Frames	The total number of frames in the dataset of vehicles
Global_Time	Time elapsed since 1970.1.1 (unit: ms)
Local_X	X-value in study regional coordinate system
Local_Y	Y-value in study regional coordinate system
Global_X	X-value in standard geographic coordinate system
Global_Y	Y-value in standard geographic coordinate system
v_length	Vehicle length (unit: feet)
v_Width	Vehicle width (unit: feet)
v_Class	Vehicle type (1: motorcycle, 2: compact vehicle, 3: large vehicle)
v_Vel	Instantaneous vehicle speed (unit: feet/s)
v_Acc	Instantaneous vehicle acceleration (unit: feet/s^2)
Lane_ID	Number of the current lane of the vehicle
Preceding	Vehicle identification number of leading vehicle
Following	Vehicle identification number of following vehicle
Space_Headway	Head spacing (unit: feet)
Time_Headway	Headway time distance (unit: s)

The acquisition interval is 0.1 s; 1 ft = 0.3048 m.

the integrity of the CF process and to ensure an adequate number of samples for model training

The pseudocode for the data processing is shown in Algorithm 2.

Table 3 is a part of vehicle tracking data obtained through data preprocessing, whose lead vehicle ID is 66 and the following vehicle ID is 74.

4.1.2. Driver Reaction Time. The driver reaction time refers to the time between the driver perceiving a change in the surrounding environment and responding [28, 29]; it is also an important parameter in the CF model.

According to the restrictions of the CF, as the driving state of the leading car changes, the following car changes accordingly. However, changes in the driving states of the two are asynchronous. This is because the driver of the following car must have a reaction process to respond to a change by the leading vehicle. This reaction process includes four parts: perception, judgment, reaction initiation, and reaction execution; the required time is referred to as the reaction time. Assuming that the reaction time is T when the leading car makes a change at time t, the following car can only make the corresponding change at time $(t + T)$.

The driver reaction time has been extensively investigated by many researchers. Kim et al. [30] analysed the braking reaction time of young and old drivers. Jin [31] studied the driver reaction time using least squares analysis via SPSS 13.0 and generated a reaction time distribution map (Figure 7). The calculation showed that the weighted average of driver reaction time is 1.077 s.

Lu et al. [32] obtained statistical information on driver reaction time through 63 samples, as shown in Table 4.

Based on the above discussion, we set the reaction time to 1 s in this study. Because the acquisition time interval of the adjacent two records of the CF data is 0.1 s, we excerpted one record for every 1 s (i.e., 10 records); the new dataset was saved for later use.

4.2. Evaluation Index. In this paper, MSE is used as the evaluation index of CFDT model. MSE is a risk function, related to the expected value of the squared error loss or quadratic loss. MSE is arguably the most important criterion used to evaluate the performance of a predictor or an estimator. It measures how close a fitted line is to data points. For every x data point, take the distance vertically from the point to the corresponding y value on the curve fit (the error), and square the value. The lower estimation value of MSE represents the lower error [33, 34]. The specific calculation formula of MSE has been shown in equation (21).

5. Training and Test

In this section, we used Keras, a Python-based deep learning library, to construct and train the CFDT model, and used TensorFlow as a back-end tool. The hardware environment of our experiment is as follows: processor Intel Xeon 2.10 GHz E5-2683 v4, memory 64 GB 2400 MHz, operating system Windows Server 2012 R2 standard, IDE: PyCharm.

According to this model training process, the optimal length of a "memory" time interval and the optimal structure

```
Processing of vehicle trajectory data.
Input: Vehicle Trajectory Data, the number of trajectory records n
Output: Car-following Data
for pos_Lv ← 0 to n
    if Following[pos_Lv] ≠ ∅ then
        time ← 0
        for pos_Fv ← 0 to n
            if Vehicle_ID [pos_Fv] == Following[pos_Lv] and Fram_ID[pos_Lv] == Fram_ID[pos_Fv] then
                time ← time + 1
                Data ← Data + [v_Vel[pos_Lv], v_Vel[pos_Fv], v_Acc[pos_Fv], Space_Headway − v_length[pos_Lv]]
        end if
    end for
    if time > 450 then
        The Data that holds the data is stored in CSV format
    end if
end for
```

Algorithm 2: The pseudocode for the CF data processing.

Table 3: A group of following team part of the data.

Lead vehicle speed	Following vehicle speed	Following vehicle acceleration	Space
22.07	18.5	0	27.72
22.07	18.5	0	28.07
22.07	18.5	0	28.41
22.22	18.5	0	28.77
22.26	18.5	0	29.17
21.94	18.5	0	29.57
21.17	18.57	1.07	29.93
20.05	18.71	1.84	30.14
19.01	18.78	0.39	30.19
18.33	18.7	−2.08	30.12
18.01	18.46	−3.41	30.03
17.96	18.19	−2.24	29.99
17.99	18.04	−0.59	30
18	18	−0.11	30
18	17.99	0.05	30
18	18	0.16	30
18	18.02	0.26	30
18	18	−0.48	29.99

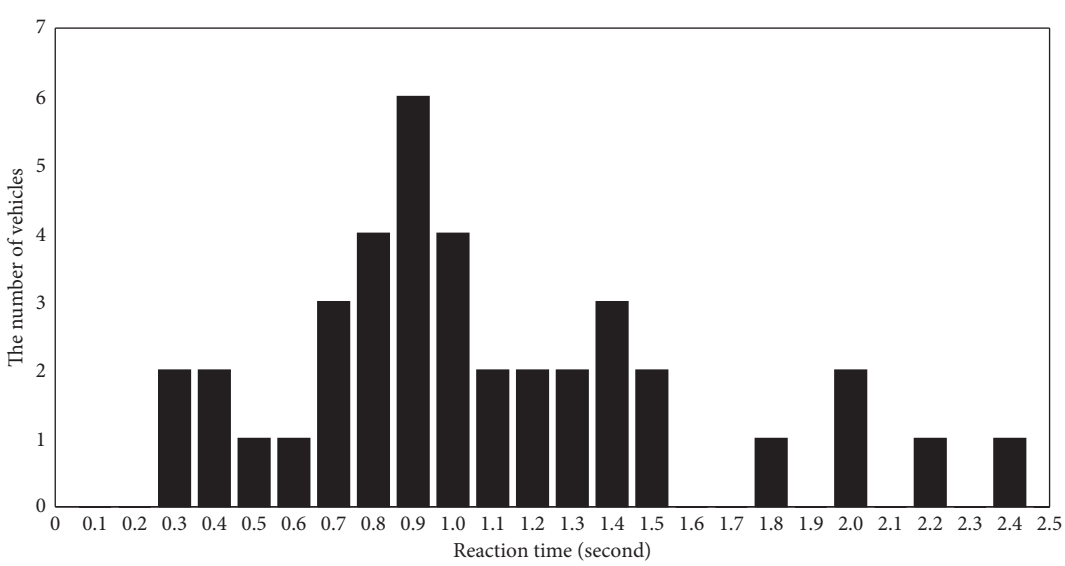


Figure 7: Reaction time distribution map.

Table 4: Reaction time statistics.

	Response time $\tau(s)$
Mean value	0.7016
Standard error	0.05236
Median	0.50
Standard deviation	0.41562
Variance	0.173
Range of change	1.70
Minimum	0.30
Maximum	2.00

of the model must be determined so the model has its best performance. The results of previous studies have shown that the optimal length of a "memory" time interval is not related to the structure of the model, and the NGSIM dataset does not need more than three hidden layers for model performance [21]. On this basis, we performed the following experiment.

First, we fixed the structure of the model. To reduce the training time, a simple structure with a single hidden layer and 20 neurons was adopted to separately perform the experiment with different N values. The results are shown in Figure 7.

Figure 8 shows that when the length of the "memory" time interval was 10, i.e., when the driver of the following car only considers the historical information within the time period of the last 10T, the model had the best performance.

On this basis, assuming $N = 10$, we separately trained and tested the models with the nine structures listed in Table 5.

The results showed that, for models with one hidden layer (Structures 1 through 3), the performance value of the model with Structure 3 is the minimum; for models with two hidden layers (Structures 4 through 6), the performance value of the model with Structure 5 is the minimum; and for models with three hidden layers (Structures 7 through 9), the performance value of the model with Structure 8 is the minimum. In terms of model performance, the performance values of the models are ranked in an ascending order: the model with Structure 3< the model with Structure 5< the model with Structure 8.

Next, the CF data, in which the vehicle ID of the leading car is 66 and that of the following car is 74, were trained using the model with Structure 3, the model with Structure 5, and the model with Structure 8. The simulation results were visualized and are shown in Figures 9–11.

In each of the figures, the first subplot shows the actual data, the second subplot shows the simulation data cluster belonging to 100 different training models, and the third subplot shows the simulation data cluster with error bars, i.e., mean ± std.

In summary, for the RNN-based CF model, the model with a "memory" time interval length of $N = 10$ and three hidden layers that contain 30, 10, and 10 neurons had the highest prediction accuracy and generated satisfactory simulation results for a road section that had continuous acceleration and deceleration behaviours.

6. Comparison with Other CF Models

To test the simulation accuracy of the RNN-based CF model, two other models, i.e., BPNN and SVR, were selected from the data-driven models to conduct a comparative experiment. Similarly, to ensure the fairness of the comparison, the speed of the leading car at time t $(v_n(t))$, the speed of the following car at time t $(v_{n+1}(t))$, and the distance between the two cars at time $(\Delta x(t))$ were used as the inputs to the model, and the acceleration of the following car at time $(a_{n+1}(t+T))$ was used as the output of the model. MSE continued to be adopted as the criterion for model evaluation.

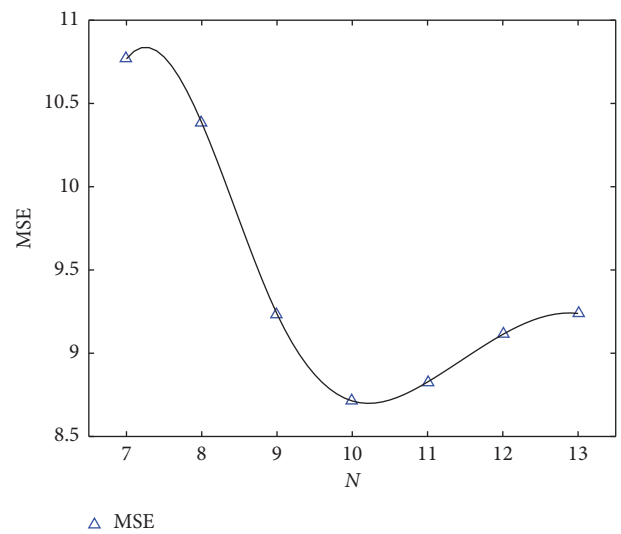


Figure 8: Influence of the length of the "memory" time interval.

Table 5: Structures of the model.

Structure number	Hidden layer 1	2	3
1	10	0	0
2	30	0	0
3	**50**	**0**	**0**
4	10	5	0
5	**30**	**10**	**0**
6	50	20	0
7	10	5	5
8	**30**	**10**	**10**
9	50	20	20

6.1. BPNN-Based CF Model. First, a model as shown in Figure 12 was constructed based on the BPNN. The model can have various structures in terms of the number of hidden layers and the number of neurons in each layer. According to Kolmogorov's theorem [35], a back-propagation neural network with three layers is sufficient to complete any mapping from n dimensions to m dimensions. Therefore, we choose the BPNN network with two hidden layers as the structure of the BPNN-based CF model. After multiple tests, the optimal structure, which included two hidden layers containing 20 and 10 neurons, was selected. The Tan h function was used as the activation function of the neurons.

The model was constructed and trained using Keras. Methods such as the holdout method were used to randomly create the training and test sets.

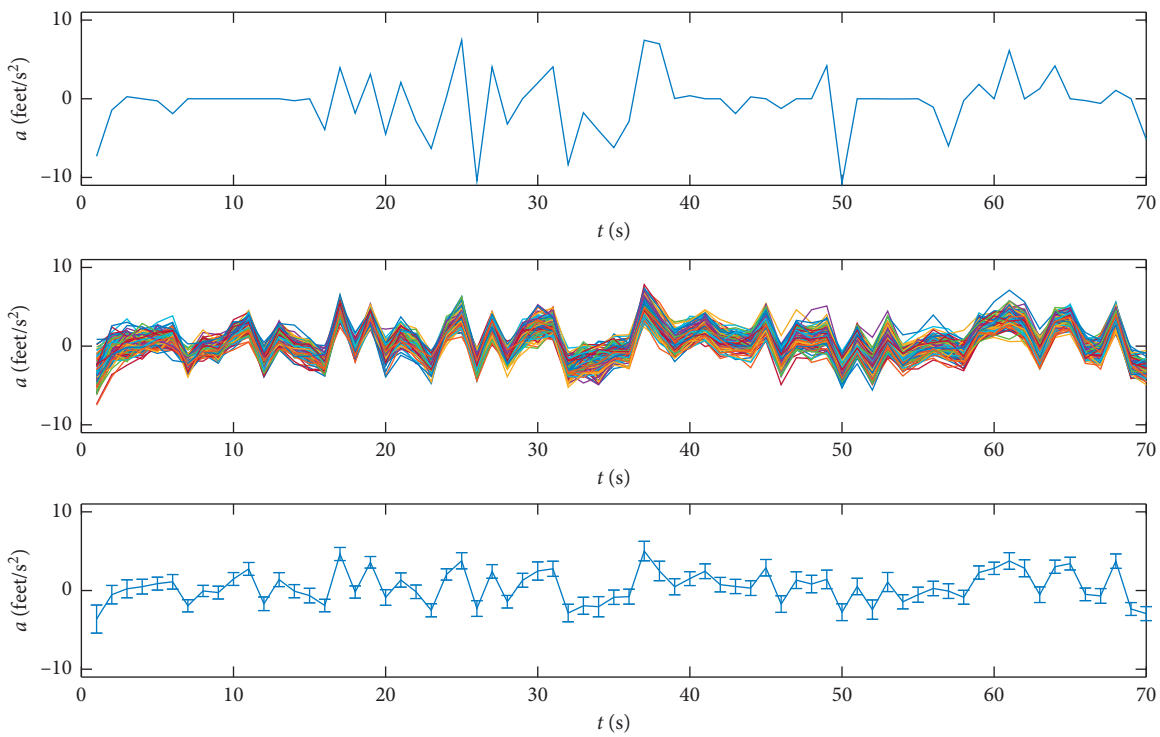


Figure 9: The model with Structure 3.

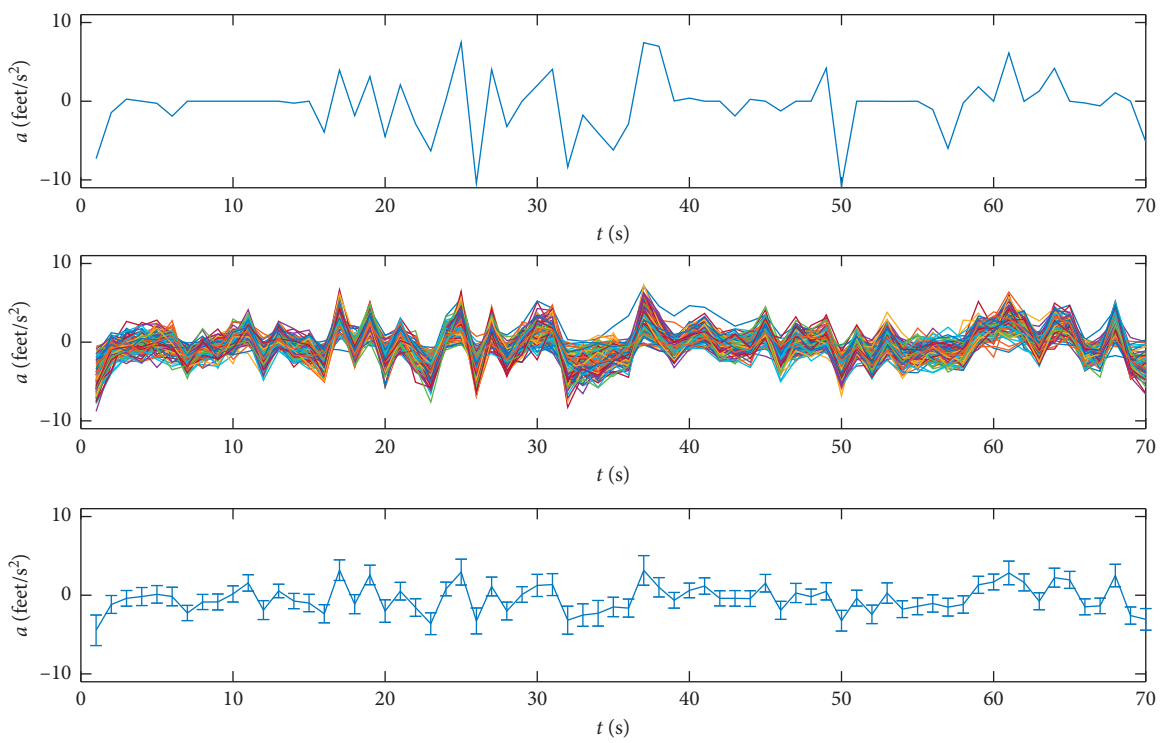


Figure 10: The model with Structure 5.

The CF data with the leading car ID of 2771 and the following car ID of 2812 were randomly selected; their simulation results are shown in Figure 13.

Figure 12 shows that because this dataset had a significant amount of "noise" (i.e., the following car was constantly in an accelerating or decelerating state), the resulting simulation results of the model have a low accuracy (i.e., there is a certain error when compared to the real data). Nevertheless, the results reflect the variation trend in the acceleration of the following car. Moreover, the model has a

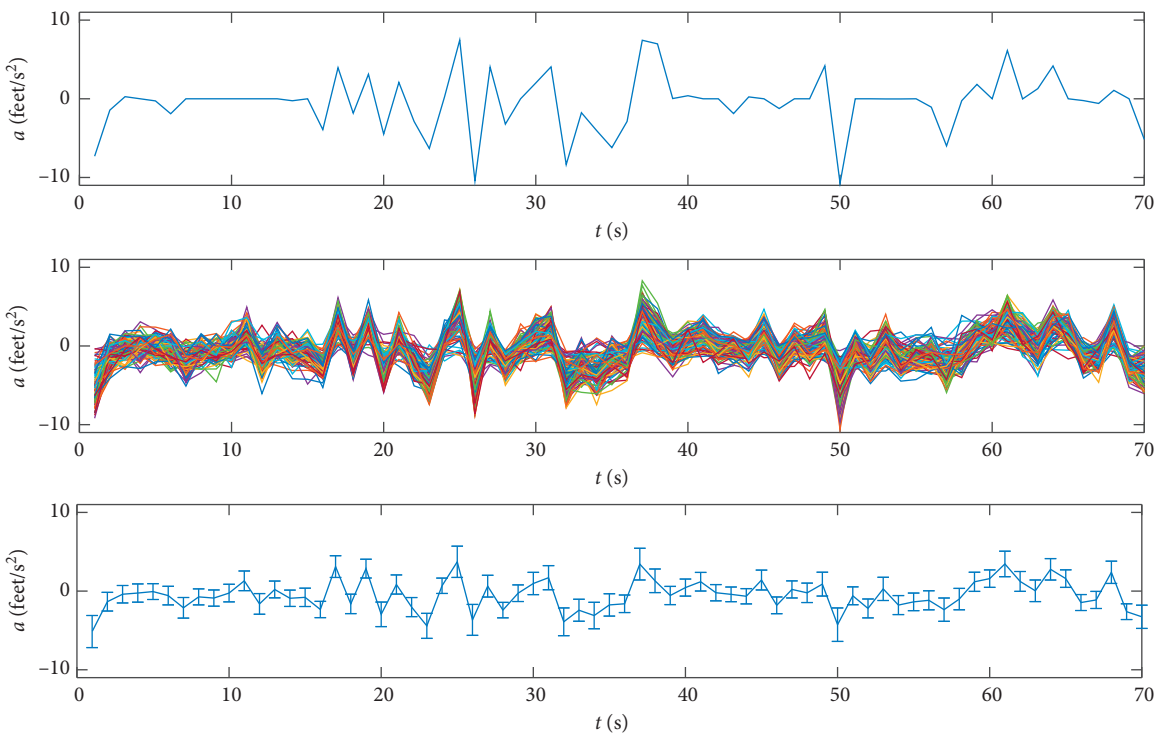


Figure 11: The model with Structure 8.

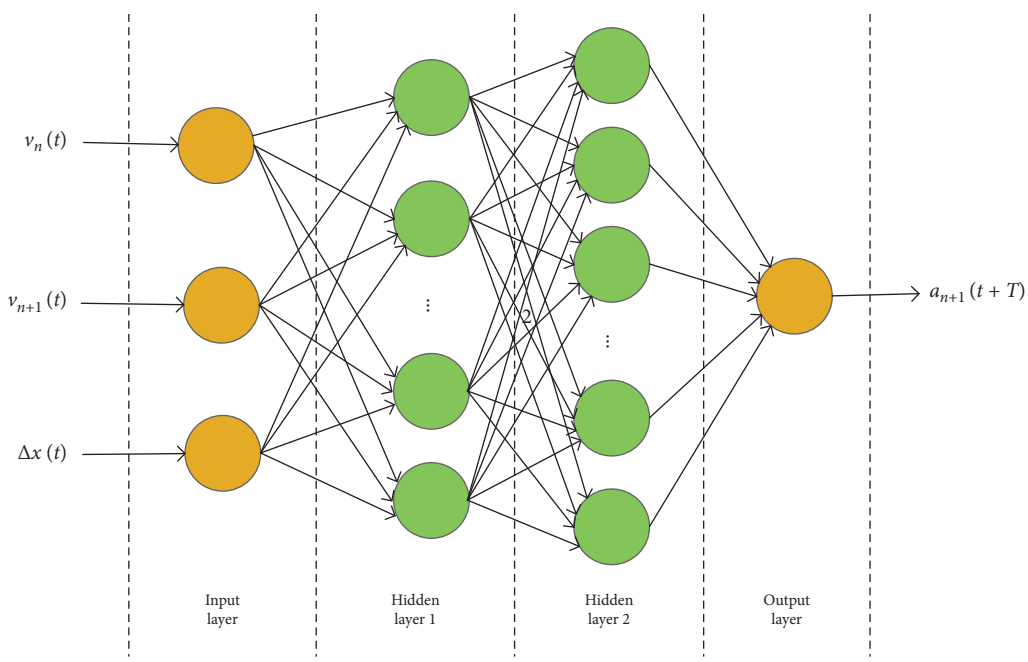


Figure 12: The BPNN-based CF model.

relatively simple structure and fast convergence rate. The above experiment demonstrates the validity of the BPNN-based CF model.

6.2. SVR-Based CF Model. we constructed an SVR-based CF model, as shown in Figure 14.

where the Gaussian kernel function was selected as the kernel function for the model.

$$K_{\text{Gaussian}}(x, z) = \exp\left(-\frac{\|x - z^2\|}{2\sigma^2}\right). \tag{22}$$

To simplify the experimental process, we used the svm. SVR in the existing Scikit-learn framework for training and

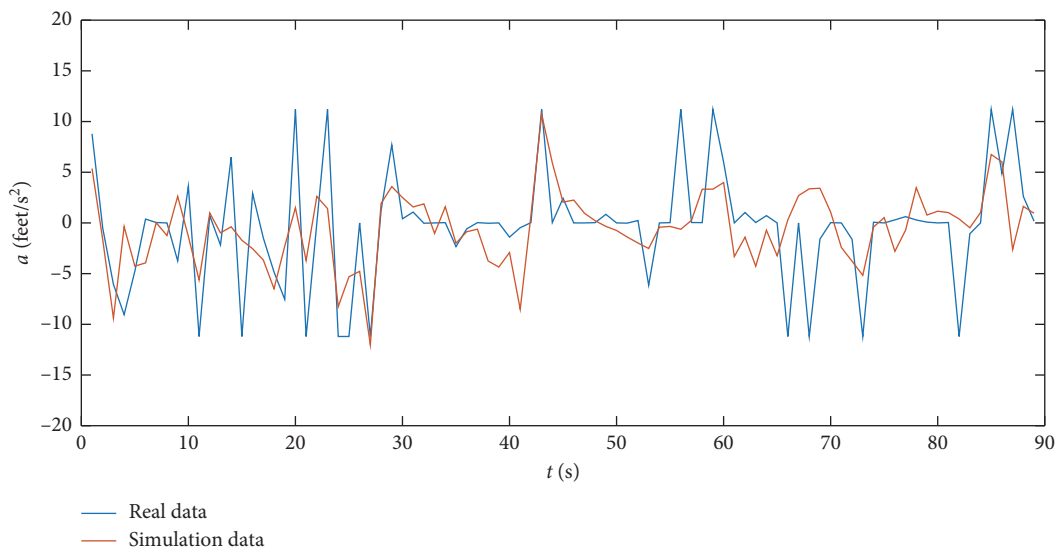


Figure 13: Simulation results of the BPNN-based CF model.

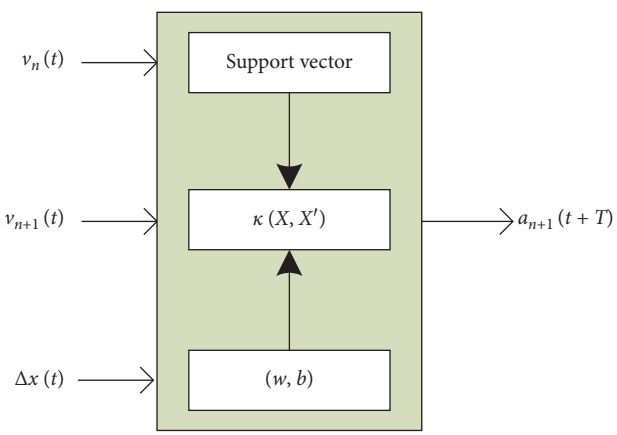


Figure 14: The SVR-based CF model.

testing and assumed the parameter kernel of svm. SVR was "rbf." As in the case of the BPNN-based model, the dataset was randomly split.

We again randomly selected a set of CF data that was associated with leading car 79 and following car 87. In addition, the following car exhibited frequent acceleration and deceleration behaviour. As shown in Figure 15, the simulation results indicate that the prediction error of the SVR-based model is small.

6.3. Comparison between Different CF Models. To compare the above three CF models, we tested each of the models on the same CF dataset (i.e., leading car 1503 and following car 1507). The simulation results are shown in Figure 16.

In this CF dataset, the acceleration and deceleration behaviour of the following car was infrequent, and the following car was driving at a constant speed nearly 40% of the time. As reflected in Figure 16, the polyline of the real data is relatively smooth. Therefore, all three models performed well in the simulation. However, in Figure 16, the GRU predicted acceleration is closer to the real acceleration than the BPNN model and the SVR model when the real acceleration of 0 to 5 seconds and 50 to 60 seconds has a significant continuous jump. Therefore, the GRU model proposed in this paper is superior to the SVR model and the BPNN model.

Figure 17 shows the MSE evaluation values of the above three models. The lower estimation value of MSE represents the lower error. Through the comparison of MSE evaluation index values in the above three model simulation experiments, it is found that the MSE value of BPNN model is the highest, while the model proposed in this paper is the lowest, which is only half of the MSE value of BPNN model. Therefore, the comparison of the MSE evaluation index values in the above three model simulation experiments shows that the proposed model is superior to the SVR model and the BPNN model.

To conclude, the above three models can not only reflect the variation trend in the acceleration of the following car but also accurately predict the values. However, for vehicles driving at variable speeds over a long period of time, the proposed model, which includes a memory unit, had better simulation results but, correspondingly, a slower convergence rate than the other two models.

7. Test Verification on Apollo Simulation Platform

The proposed method for predicting a car-following behaviour is combined with the Apollo platform as follows:

Differentiating scene information in an autonomous driving process of a vehicle into static information and dynamic information, and importing the static information and the dynamic information into Dreamview of the Apollo platform to construct a road scene, specifically including obtaining three-dimensional information of a traffic scene

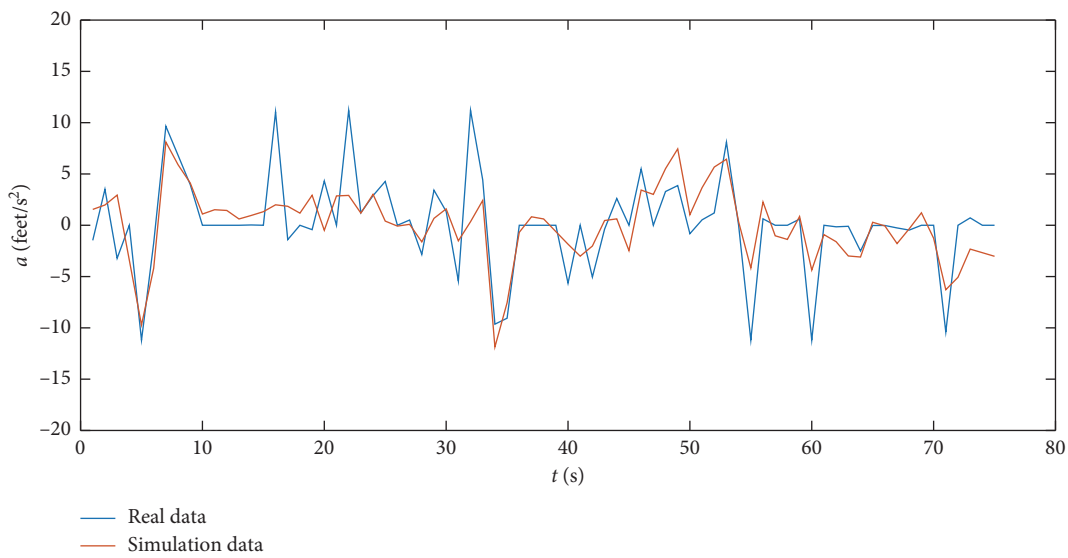


Figure 15: Simulation results of the SVR-based CF model.

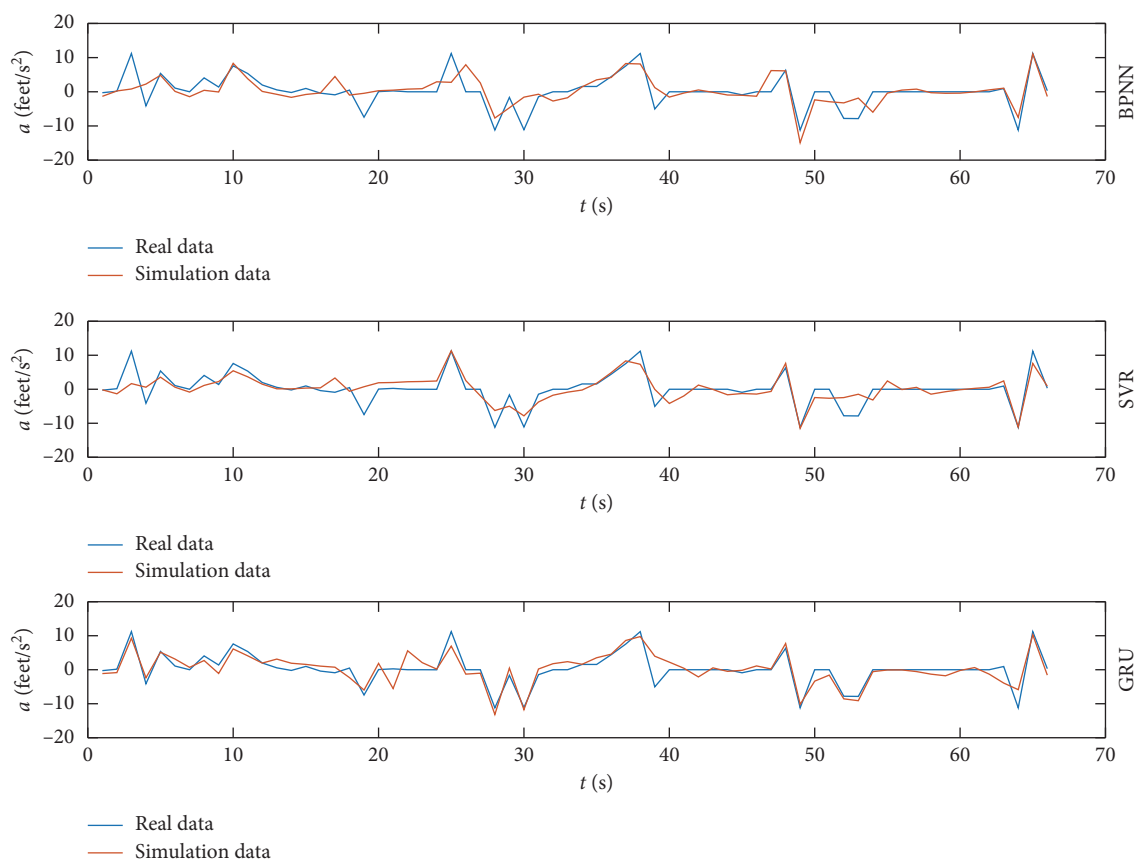


Figure 16: Comparison of the simulation results of the three CF models.

and motion information, wherein the three-dimensional information of the traffic scene refers to static information in the corresponding scene information and the motion information of the traffic scene refers to dynamic information in the scene information; preliminarily constructing a topological structure of the scene, wherein the topological information of the scene includes information such as the number of surrounding vehicles, the lanes occupied by surrounding vehicles, and the distance from a road edge; inputting such information into Dreamview via a corresponding interface of Apollo; configuring paths to specific modules based on the Table of Module Output Interface

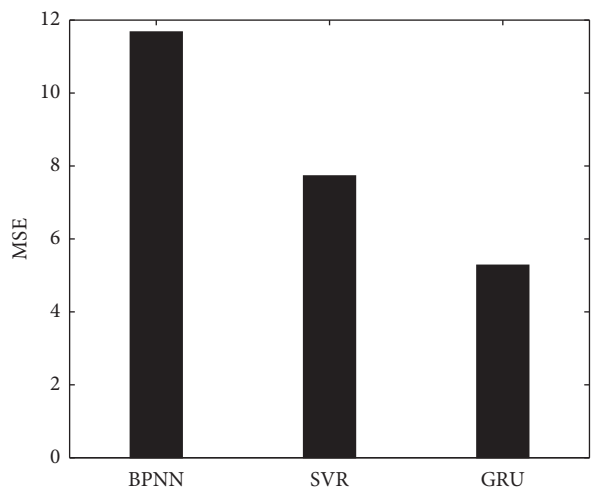


Figure 17: Comparison of the MSE evaluation values of the three CF models.

Table 6: Table of module output interface standards provided by simulation.

Module	Topic	Description	Field
Localization	/apollo/localization/ pose	Output data such as the position and orientation of the following car	Position, orientation, linear_velocity, linear_acceleration, angular_velocity in pose
Perception	/apollo/perception/ obstacles	Output data such as positions, orientations, velocities, shapes, etc. of respective obstacles	The simulation will provide id, position, theta, velocity, length, width, height, type, polygon_point in PerceptionObstacle
	/apollo/perception/ traffic_light	Output traffic light signals	The simulation will provide color, id and tracking_time in Traffic Light,
CAN bus	/apollo/canbus/ chassis	Output data such as the velocity and drive mode of the following car	The simulation will provide speed_mps
Router	/apollo/ routing_response	Output the navigation result	The simulation will provide routing response, including the planned navigation route from the current position to the destination

Standards (Table 6) provided by the simulation environment, and performing, by respective modules in the Standard, environment construction with reference to the traffic flow and simulated environment information resulting from understanding of the scene, as shown in Table 6.

During the construction process, the dynamic information and the static information during vehicle driving are obtained through understanding of the scene; the desired distance and reaction time are obtained by capturing the driver's behaviour features, and the process of the car-following model is improved using RNN, computing the safety-and-comfort-based optimal solution of the following-car acceleration range. Meanwhile, the model is tested and validated using the Apollo simulation platform to ensure accuracy and utility of the model.

The method for predicting a car-following behaviour under the Apollo platform as disclosed is tested using the Apollo simulation platform. After the Apollo software environment is configured, the output interface of the Apollo platform is docked with the method. After successfully predicting information such as the following-car acceleration using the proposed method, the method is docked with the decision planning module Planning of Apollo; finally, the Apollo software implements testing and validation of the method; during multiple times of simulation process, the parameters are constantly adjusted, and the algorithm is optimized over the Apollo visual platform, specifically:

Deploying the environment (e.g., Docker environment), and pulling the container mirror of Apollo; entering the Apollo container, and compiling the simulation environment (e.g., Dreamview simulation environment); running the simulation environment after successful compilation; testing and validating the efficacy of the model using the corresponding simulation environment; the testing and validating interface refers to the simulation environment interface, wherein the interface is shown in Figure 18.

Docking the traffic flow and environment information outputted by Apollo with the input of the model; then, converting the predicted acceleration value obtained using the model into the Planning input simulation platform, wherein the specific docking path is shown in Table 7.

Docking the traffic flow and environment information outputted by Apollo with the input of the model; then, converting the predicted acceleration value obtained through the model into the Planning input simulation platform, wherein the specific docking path is shown in Figure 19.

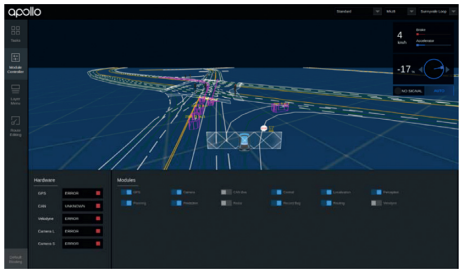


Figure 18: A simulation Apollo platform.

Table 7: Table of decision planning module interface standards.

Module	Topic	Description	Fields
Planning	/apollo/ planning	Output the planned trajectory of the following car in a future period of time	The developer must provide (1) timestamp_sec in header (2) v, a, relative_time in trajectoryPoint (3) x, y, z, theta, kappa in pathPoint
Prediction	/apollo/ prediction	Output respective obstacles and their predicted trajectories	The developer optionally outputs trajectories in predictionObstacle, which may be used for displaying predicted trajectories of respective obstacles
Decision	/apollo/ decision	Output decisions with respect to various obstacles and the main decisions	The developer optionally outputs mainDecision and objectDecisions, which may be used for displaying the main decisions and the decisions with respect to respective obstacles

Testing and validating the model in the Apollo simulation environment.

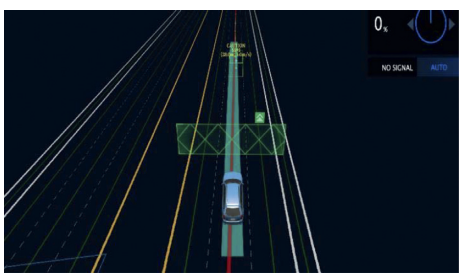


Figure 19: Simulation environment.

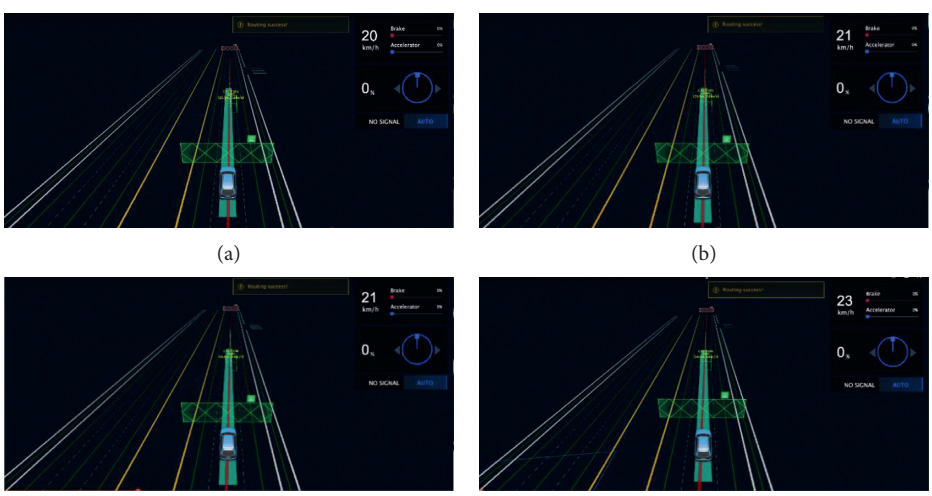


Figure 20: The following process under Apollo platform.

Figure 20 is part of the following process diagram of the vehicle following behaviour visualization on the Apollo platform using the model proposed in this paper. In the figure, the cube with green border is the leading car and its current speed is shown directly above it. The blue car is the following car and its current speed is shown at the top right of the figure. In Figure 20(a), the speed of the leading car is 5 m/s (equivalent to 18 km/s), and the speed of the following car is 20 km/s. In Figure 20(b), the speed of the leading car is 5 m/s (equivalent to 18 km/s), and the speed of the following car is 21 km/s. In Figure 20(c), the speed of the leading car is 5 m/s (equivalent to 18 km/s), and the speed of the following car is 21 km/s. In Figure 20(d), the speed of the leading car is 5 m/s (equivalent to 18 km/s), and the speed of the following car is 23 km/s. In the comparison of Figures 20(a)–20(d), it is found that the speed of the following car has been fluctuating from the speed of the leading car so that the speed of the following car is not longer than that of the leading car. Moreover, a relatively long distance is maintained between the following car and the leading car, which provides sufficient braking distance for the following car to ensure the driving safety of the following car. Through the visual simulation of the model proposed in this paper on the Apollo simulation platform, it was found that there would be no collision between the following car and the guiding car during the long time of following. Therefore, by using Apollo simulation platform to test the model visually, it is found that the model proposed in this paper is valid and practical.

8. Conclusions

In this study, we used high-precision vehicle trajectory data from Interstate I-80 in the NGSIM dataset to obtain the CF data through data preprocessing. Based on the characteristics of the CF behaviour, we verified the correctness of the data through experiments and chose the reaction time of $T = 1$ s to further filter the CF data. We modelled the CF behaviour based on the RNN network, using the speed of the leading car, the speed of the following car, and the distance between the two cars, all at time t, as inputs and the acceleration of the following car at time $(t + T)$ as output. Finally, we performed in-depth examination and experiments on the length of the "memory" time interval and the network structure of the constructed model and obtained the parameters that enabled the model to have its best performance. Further, we compared the simulation results of the constructed model with those of the BPNN- and SVR-based models and demonstrated that the constructed model with a memory unit added had higher simulation accuracy. Both the BPNN- and SVR-based CF models were unstable; i.e., when the following vehicle had frequent acceleration and deceleration behaviour, their simulation results were poor. In comparison, the RNN-based CF model was able to make more accurate predictions because it considered the relevant information of the last several time intervals. At the same time, Apollo simulation platform was used to test and verify the model, ensuring the accuracy and practicability of the model.

The RNN-based CF model established in this study was only used to study the CF behaviour between small cars. In reality, the CF behaviours between different types of vehicles are different. In a follow-up study, we will construct models that consider various types of vehicle so that the existing model can be further improved according to the actual conditions such as asymmetric driving behaviour and multiple leading vehicles.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge the support from CERNET Innovation Project under Grant NGII20161201, Scientific and Technological Planning Project of Beilin District of Xi'an under Grant GX1819 supported by National Key Research and Development Program of China (2018YFB1201500), Natural Science Foundation of China (no. 61773313), and Teaching Research Foundation of Xi'an University of Technology (XJY1866).

References

[1] B. G. Cao, "A car-following dynamic model with headway memory and evolution trend," *Physica A: Statistical Mechanics and Its Applications*, vol. 539, Article ID 122903, 2020.

[2] G. H. Peng, S. H. Yang, D. X. Xia, and X. Q. Li, "Delayed-feedback control in a car-following model with the combination of V2V communication," *Physica A: Statistical Mechanics and Its Applications*, vol. 526, Article ID 120912, 2019.

[3] L. Zheng, C. Zhu, Z. He, T. He, and S. Liu, "Empirical validation of vehicle type-dependent car-following heterogeneity from micro- and macro-viewpoints," *Transportmetrica B: Transport Dynamics*, vol. 7, no. 1, pp. 765–787, 2018.

[4] L. Bin, *Modeling and Analysis of Car-Following Behavior Using Data-Driven Methods*, Southwest Jiaotong University, Chengdu, China, 2017.

[5] L. Li, D. Zhang, Z.-G. Xu, P. Wang, and G.-P. Wang, "The roles of car following and lane changing drivers' anticipations during vehicle inserting process: a structural equation model approach," *Journal of Advanced Transportation*, vol. 2018, Article ID 6372861, 19 pages, 2018.

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 696–699, 1988.

[7] C. H. Chen, F. Y. Song, F. J. Hwang, and L. Wu, "A probability density function generator based on neural networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 541, Article ID 123344, 2020.

[8] N. Kehtarnavaz, N. Groswold, K. Miller, and P. Lascoe, "A transportable neural-network approach to autonomous vehicle following," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 2, pp. 694–702, 1998.

[9] J. H. Zhang, Q. Li, and D. P. Chen, "Safety assistance algorithm for longitudinal collision avoidance based on BP neural network," *Journal of Xi'an Jiaotong University*, vol. 51, no. 7, pp. 140–147, 2017.

[10] B. Lu, N. Shaoquan, and S. S. Washburn, "A support vector regression approach for investigating multianticipative driving behavior," *Mathematical Problems in Engineering*, vol. 2015, Article ID 701926, 10 pages, 2015.

[11] D. Wei and H. Liu, "Analysis of asymmetric driving behavior using a self-learning approach," *Transportation Research Part B: Methodological*, vol. 47, pp. 1–14, 2013.

[12] P. Parham, M. P. A. Moghadam, and B. Bigdeli, "Car following prediction based on support vector regression and multi-adaptive regression spline by considering instantaneous reaction time," *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, vol. 43, no. S1, pp. 67–79, 2019.

[13] X. L. Huang, J. Sun, and J. Sun, "A car-following model considering asymmetric driving behavior based on long short-term memory neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 346–362, 2018.

[14] X. Zhang, J. Sun, X. Qi, and J. Sun, "Simultaneous modeling of car-following and lane-changing behaviors using deep learning," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 287–304, 2019.

[15] Y. K. Wu, H. C. Tan, X. X. Chen, and B. Ran, "Memory, attention and prediction: a deep learning architecture for car-following," *Transportmetrica B: Transport Dynamics*, vol. 7, no. 1, pp. 1553–1571, 2019.

[16] C. Hua, "A new car-following model considering recurrent neural network," *International Journal of Modern Physics B*, vol. 33, no. 26, 2019.

[17] D. Yang, D. Wu, Y. Liu, and B. Ran, "A novel car-following control model combining machine learning and kinematics models for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 1991–2000, 2019.

[18] Y. Li, X. Lu, C. Ren, and H. Zhao, "Fusion modeling method of car-following characteristics," *IEEE Access*, vol. 7, pp. 162778–162785, 2019.

[19] D.-F. Xie, Z.-Z. Fang, B. Jia, and Z. He, "A data-driven lane-changing model based on deep learning," *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 41–60, 2019.

[20] B. Xiao, Y. G. Liu, and B. Xiao, "Accurate state-of-charge estimation approach for lithium-ion batteries by gated recurrent unit with ensemble optimizer," *IEEE Access*, vol. 7, pp. 54192–54202, 2019.

[21] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng, and F.-Y. Wang, "Capturing car-following behaviors by deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 910–920, 2018.

[22] J. Wang, F. Sun, and R. Cheng, "An extended car-following model considering the self-stabilizing driving behavior of headway," *Physica A Statistical Mechanics & Its Applications*, vol. 507, pp. 347–357, 2018.

[23] C.-H. Chen, "A cell probe-based method for vehicle speed estimation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E103.A, no. 1, pp. 265–267, 2020.

[24] C. J. Ou, S. M. Bedawi, A. B. Koesdwiady, and F. Karray, "Predicting steering actions for self-driving cars through deep learning," in *Proceedings of the 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Chicago, IL, USA, August 2018.

[25] J. Webber, N. Suga, A. Y. Mehbodni, K. Yano, and T. Kumagai, "Study on fading prediction for automated guided vehicle using probabilistic neural network," in *Proceedings of the 2018 Asia-Pacific Microwave Conference (APMC)*, Kyoto, Japan, November 2018.

[26] A. Chernikova, A. Oprea, C. Nita-Rotaru, and B. G. Kim, "Are self-driving cars secure? Evasion attacks against deep neural networks for steering angle prediction," in *Proceedings of the 2019 IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, USA, May 2019.

[27] FHWA, *Next Generation Simulation (NGSIM)*, FHWA, Washington, DC, USA, 2017.

[28] N. Arbabzadeh, M. Jafari, M. Jalayer, S. Jiang, and M. Kharbeche, "A hybrid approach for identifying factors affecting driver reaction time using naturalistic driving data," *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 107–124, 2019.

[29] Y. Wang, J. Zhang, and G. Lu, "Influence of driving behaviors on the stability in car following," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1081–1098, 2019.

[30] C. K. Kim, B. Yacoubi, and E. A. Christou, "Visual load and variability of Muscle activation: effects on reactive driving of older adults," *Human Movement Science*, vol. 63, pp. 172–181, 2019.

[31] Q. Jin, *Reasearch on Parameters Calibration and Verification of Car-Following Models*, Shanghai Jiao Tong University, Shanghai, China, 2008.

[32] G. Lu, B. Cheng, Y. Wang, and Q. Lin, "A car-following model based on quantified homeostatic risk perception," *Mathematical Problems in Engineering*, vol. 2013, Article ID 408756, 13 pages, 2013.

[33] A. Kumar, A. Prakash, S. Sharma, and K. Jyoti, "Vehicle authentication and message hiding protocol for vehicle to vehicle communication," in *Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, IEEE, Dehradun, India, September 2015.

[34] N. F. Mustamin, "Relative distance measurement between moving vehicles for manless driving," in *Proceedings of the 2017 International Seminar on Application for Technology of Information and Communication*, Semarang, Indonesia, October 2017.

[35] V. Kurková, "Kolmogorov's theorem and multilayer neural networks," *Neural Networks*, vol. 5, no. 3, pp. 501–506, 1992.

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 5048629, 9 pages
https://doi.org/10.1155/2020/5048629

Research Article

An Improved Swarm Intelligence Algorithm for Multi-Item Joint Ordering Strategy of Cruise Ship Supply

Liling Huang and Jiaqi Yang

School of Transportation, Wuhan University of Technology, Wuhan 430064, China

Correspondence should be addressed to Liling Huang; 316536524@qq.com

Received 7 December 2019; Revised 8 February 2020; Accepted 21 February 2020; Published 14 March 2020

Guest Editor: Chi-Hua Chen

From the perspective of the global value chain of the cruise industry, the output value of cruise operation accounts for 50%, with the highest added value, and cruise ship supply is the crucial link in cruise operation management, with substantial economic benefits. Therefore, optimizing the purchasing process can not only save costs for cruise companies but also improve cruise level service. Aiming at the purchasing characteristics and modes of cruise ship supply, an optimization model of multi-item joint ordering is constructed under global cruise ship supply chain, considering different order cycles, integrated operation of purchase, delivery, and inventory based on a cruise distribution center. And an improved swarm intelligence algorithm, called fireworks algorithm with inertia weight (WFWA), is proposed for global optimization of the objective function. By comparing the optimization results with fireworks algorithm (FWA), genetic algorithm (GA), and particle swarm optimization (PSO) through experimental tests, it demonstrates that WFWA has higher optimization accuracy and better global convergence.

1. Introduction

In recent years, the cruise industry has developed from a narrow cruise reception economy to a broad cruise industry chain, further providing policy innovation for the development of related industries such as cruise supply chain, cruise ship supply, and supporting facilities of cruise home port. In the cruise industry chain, a large part is the cruise ship supply, which is the main component of the cruise operation management and accounts for 50% of the output value, with the highest added value. At the same time, it is also an essential part of the cruise industry chain, an important indicator to measure the competitiveness of a cruise home port, and one of the key elements to attract international cruise ships to come to ports for regular operation. Although the number of cruise ports tourists is increasing year by year in China, few of them are profitable, and the profit point is concentrated in the procurement and logistics business of cooperative enterprises, without giving full play to the added value of the tourism industry. Therefore, the operation of cruise ships directly affects the economic benefits of cruise companies and plays a vital role in the economy of cruise home port.

The cruise ship is a special kind of vessel for international navigation. In the broad sense, cruise ship supplies are basically the same as ship materials, which can divided into fuel, materials, consumables, and duty-free goods. In the narrow sense, cruise materials refer to the food and daily necessities used and consumed by cruise service personnel and passengers during the voyage. For example, a cruise ship carrying 2700 passengers, which has a one-week journey (7 days and 6 nights), needs to prepare supplies including 12,000 liters of soda water, 4500 kilograms of chicken, 71,000 eggs, 2500 kilograms of bacon, 3500 liters of ice cream, 20,000 bottles of drinking water, and 10,000 bottles of liquor and beverages. Therefore, the replenishment of cruise ship supplies is enormous. Furthermore, with the large-scale development of cruise ships, ship tonnage and passenger capacity gradually increase, which bring many challenges to cruise ship operation management [1]. Therefore, the ordering of cruise ship supply needs to make full use of the advantages of globalization to achieve cost reduction. Cruise

companies should focus on solving and ensuring the uninterrupted cruise ship supply during the voyage. Moreover, the multi-item joint ordering plays a role in cruise supply chain management and coordination, and the cruise distribution center is an important part of the cruise ship supply logistics system as the final hub of ship supplies.

Compared with the supply of ordinary cargo ships, the purchase of cruise ship supplies has the following characteristics:

(1) Global procurement: compared with ordinary cargo ships, cruise ships carry a large number of passengers. For example, there are more than 8000 people in the Quantum of the Seas which needs a large number of living materials and other spare parts. To ensure quality and reduce procurement cost, cruise ship supplies require global sourcing.

(2) High frequency replenishment: cruise ships usually have a period of several days or weeks to realize the cycle of departure and return. Compared with freighters, materials are consumed quickly and there is a need for replenishment in a short time. Besides, cruise ship needs consumable replenishment when it arrives at an intermediate location every time.

(3) The cruise company purchases directly from suppliers, while the cargo ships purchase from agent companies.

(4) A centralized hub is needed to check the quality and safety of the supplies, which ensures that supplies arrive at the cruise ships within the expected time.

The remainder of the paper is as follows. In Section 2, a relevant literature review is provided. In Section 3, the problem is formally discussed and an optimization model is formulated. Later, Section 4 proposes an improved swarm intelligence algorithm to solve the optimization problem. Afterward, the results and analysis are presented compared with other three swarm intelligence algorithms through a case study in Section 5. Eventually, Section 6 concludes this article and suggests possible research directions in the future.

2. Literature Review

2.1. Cruise Ship Supply. Jones [2] pointed out that ship replenishment materials provided by ship suppliers mainly include dry goods, food, cabin supplies, deck supplies, tools supplies, bonded materials, and oil materials. Véronneau and Roy [3] analyzed the planning process of the cruise supply chain and proposed that the cruise operation focuses on logistics supply. Véronneau and Roy [3] and Vaggelas and Lagoudis [4] considered that the global cruise supply chain faced the challenge of supply point mobility, and the replenishment ports were always under dynamic change. Moreover, with the distance between the port of call and the raw material supply point farther away, the purchase lead time further increased. According to Boyd tactical cycle theory, Véronneau and Cimon [5] proposed that the purchase lead time should be shorter than cruise length to remain the responsive within a certain period, and shortening the delivery time was the primary solution to provide more flexibility and further shift to a pull supply system. Lois et al. [6] considered that a holistic and systematic approach was necessary to ensure the safety and security of the cruise ship, so cruise ship supply needed to flow through a centralized distribution center to check quality and safety and ensure that supplies arrive onboard as expected. Compared with the rapid development of cruise tourism in China, the research on cruise ship supply is relatively lagging, and most of them are macroscopic. For example, Fu and Shou [7] and Liu [8] proposed that cruise ship supply chain was complex, global, time-sensitive, irreversible, and subject to supply variability. Yin [9] studied the distribution problem of cruise ship supply, which had the characteristics of large quantity, high value, and strict time requirement and required specialized cruise distribution centers and supporting facilities to centralized procurement, centralized supervision, centralized distribution, and centralized settlement for the distribution development. Zheng et al. [10] studied the fuel consumption of a voyage for cruise ships.

Based on the above analysis, more and more scholars pay attention to the replenishment of cruise ship supply. Because each material has different order cycle in multisourcing mode for cruise ship supply, a minimal cost model of cruise ship supply logistics is established from the perspective of cruise ship supply chain management, combined with the characteristics and purchasing mode of cruise ship supply logistics, considering purchase price, order cycle, inventory capacity, etc. In the existing studies, few of them have attempted to solve a similar problem.

2.2. Joint Ordering. Balintfy [11] was the first to propose the (S, c, S) strategy, marking the beginning of the research on multi-item joint ordering; this system was a particular continuous inventory system used to coordinate the order of goods. Goyal [12] and Sivler [13] studied the multi-item material inventory model without considering the constraints of capital, resources, and inventory capacity, assuming that the demand rate was constant, the order price was fixed, and there was no shortage. Pirkul and Aras [14] proposed a multiproduct inventory model considering constraints of single resource capital, warehouse volume, and service level. Domestic scholars focused on the uniform ordering strategy and joint ordering strategy with fixed cycle. For example, Xu and Xu [15] studied the method of joint ordering in batches and fixed a minimum order cycle as the basic order cycle; others are integer times of the basic order cycle, and the ordering quantity was determined by respective demand rate, recovery rate, repair rate, and order cycle. Zhu et al. [16] analyzed the advantages, disadvantages, and conditions of the aggregation effect for three strategies: individual ordering, joint ordering, and unified ordering for multi-item or multiple customer ordering strategy in the supply chain. Wen and Da [17] established a multi-item inventory model for a deteriorated product supply chain and analyzed the properties of the total cost function based on

five different ordering strategies. Besides, in the research of multi-item joint ordering optimization method with stochastic demand, Zhang et al. [18] discussed the joint ordering inventory problem based on random continuous inventory strategy considering service level constraints, and a heuristic algorithm for key decision parameters was proposed for the multi-item joint ordering problem. Zhang et al. [19] built a multi-item joint ordering model based on fixed-period strategy under the condition of stochastic demand, taking into account the constraints of guarantee degree, using the Monte Carlo stochastic simulation technology to solve the expected value of random variables, and genetic algorithm was used to calculate the minimum inventory cost of the basic order cycle, order cycle of each product, and the maximum inventory. Guan and Wu [20] established a fuzzy multiobjective mixed-integer programming model of single-stage and multi-item ordering quota allocation with fuzzy demand and price discount, and the two-stage algorithm was applied to solve it.

At present, the research on joint ordering problem and algorithms has made great achievement [21–26]; most of them adopt heuristic algorithms to solve it, such as genetic algorithm (GA), particle swarm optimization (PSO), differential evolution algorithm (DE), and improved or hybrid algorithms. Furthermore, heuristic algorithms are combined with simulation methods. The inventory modeling and solving of multi-item joint ordering mainly adopt probability, mathematical programming, numerical calculation method, and optimization method. According to the characteristics of demand and other information, a multi-item joint ordering strategy can be divided into two categories: deterministic type and random type. However, there are flaws, and the cost model considered more expenses and less service level; almost all inventory cost models take "average order satisfaction rate" as the standard to measure service quality while ignoring order response time. In addition, the optimization of multi-item joint ordering problem usually only considered the minimum inventory cost and order cost, but ignored the distribution cost, or it is considered as a constant, so the joint optimization of purchase, inventory, and delivery should be considered later.

This study has several contributions to the field of operation management in the cruise ship supply logistics. First, we put forward a new thought, which is entirely consistent with multiple sourcing according to the characteristics of cruise ship supplies, and also fill the research blank of joint ordering strategy of supplies in cruise operations. Second, an improved swarm intelligence algorithm, called fireworks algorithm with inertia weight (WFWA), is used to fix cruise ship supplies problem, which is an approach to solve an optimization model for joint purchase, inventory, and delivery under the supply chain environment.

3. Problem Description and Modeling

The main supply chain of the cruise company includes four main procurement processes: fuel purchasing, corporate purchasing, technical purchasing, and hotel purchasing. Fuel purchasing is responsible for acquiring and delivering fuel and other petroleum products such as lubricants to ship for its daily consumption. Corporate purchasing looks after most items required shore side as well as some corporate items used onboard such as office supplies and computers. Technical purchasing is the item mostly used by facility departments and marine departments such as engine parts and carpets. Finally, hotel purchasing is the department ensuring that foods and other consumable items required for hotel operations are in place; the procurement model of cruise ships is shown in Figure 1.

It needs a cruise distribution center as the supplies centralized hub for centralized purchasing, supervision, delivery, and settlement in the procurement model. The cruise distribution center could store all supplies with different order cycles and deliver to loading port through item checking, system receiving, and system shipping, thus avoiding loading port congestion caused by separate distribution from different suppliers. The purchasing model is useful to reduce distribution costs, integrate resources, and improve delivering efficiency. Therefore, it is of practical significance to study the optimization of multi-item joint ordering problem for cruise ship supply based on the cruise distribution center to reduce the cost of cruise logistics.

3.1. Problem Description. With the globalization of cruise ship supplies procurement, the cruise distribution center is required as the logistics base for multi-item and multi-sourcing in the cruise ship supply chain, which has storage function. The purchase and distribution adopt inventory coordination mode, and the optimization objective is cost minimization. The system is composed of a cruise distribution center and multiple ship suppliers with import and export business qualifications, either as partner suppliers or other warehouses around the logistics base. Many suppliers provide ship supplies with different lead times, which are concentrated in the cruise distribution center, and delivery to loading port after processing, as shown in Figure 2. Therefore, there exists a multi-item joint ordering problem of cruise ship supply in cruise operations. And how to determine the order cycle and order quantity of each item, and minimize ordering cost within the demand cycle, which is the multi-item joint ordering strategy of cruise ship supply.

The following assumptions are considered in our models:

(1) The initial inventory is 0, replenishment is instantaneous, and there is no shortage. And the inventory capacity is V (constant).

(2) There is a basic order cycle for cruise ship supplies. The order cycle of all supplies is an integer multiple of the basic order cycle, and the order cycle of one material is at least equal to the basic order cycle.

(3) The basic ordering cost of various supplies is the same, but the individual ordering costs vary. The

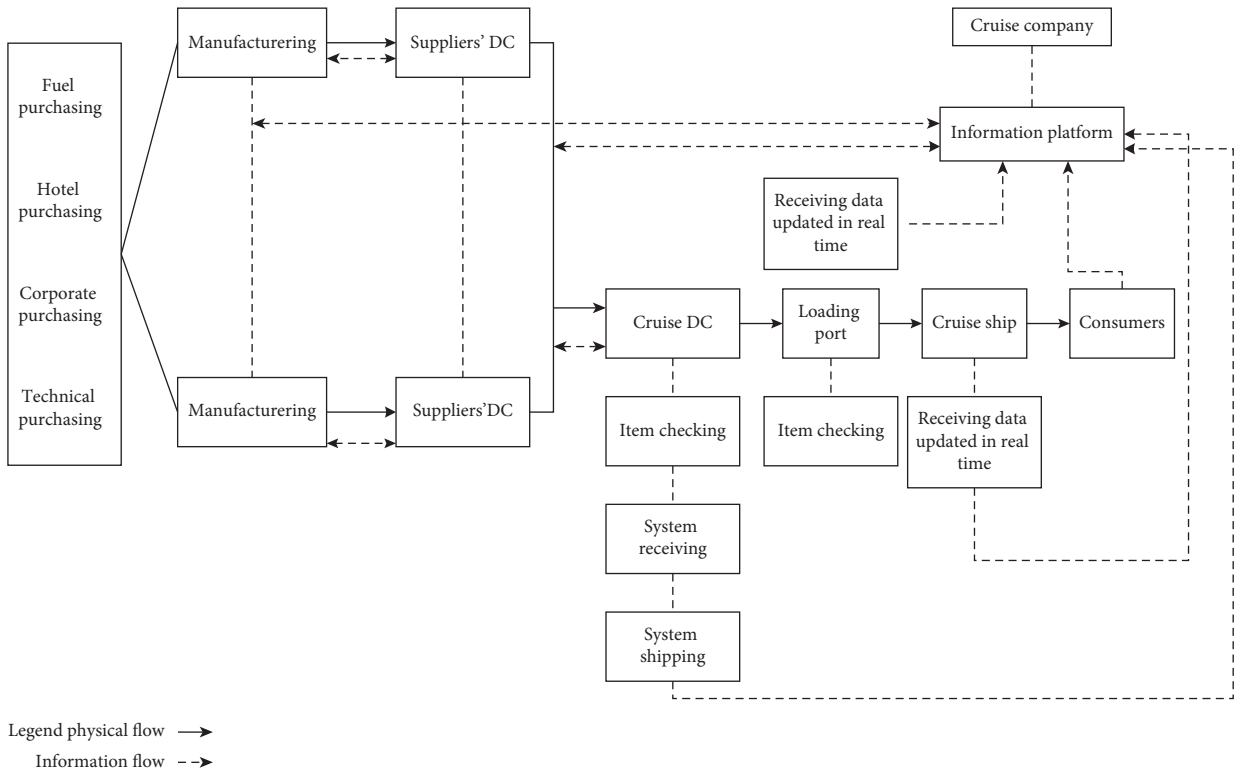


Figure 1: Cruise ship supplies purchasing model in supply chain environment.

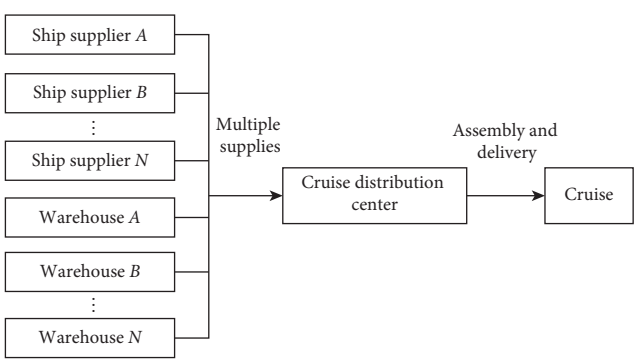


Figure 2: Joint ordering and distribution of multiple supplies based on a cruise distribution center.

subscription costs are related to the ordering times, and there is a basic subscription cost for each order.

(4) Based on the cruise distribution center, cruise companies send orders to suppliers on demand.

Definitions of parameters in our study are shown in Table 1.

3.2. Model Formulation. Supposing that the basic order cycle is t, the order cycle of material 1 is the same as material 2, which is the basic order cycle. Thus, material 1 and material 2 are divided into one group. Material 3 is twice the basic order cycle, that is, $2t$, and material 3 is divided into one group, and so on. According to the above statement, each material order cycle T_i is integer times the basic order cycle, which can formulated as follows:

$$T_i = X_i * t. \tag{1}$$

Therefore, the order quantity of material i is denoted as

$$Q_i = R_i * T_i = R_i * X_i * t. \tag{2}$$

Since the initial inventory of each material is 0, the average inventory level of material i is expressed as follows:

$$\overline{I_i} = \frac{Q_i}{2} = \frac{R_i X_i t}{2}. \tag{3}$$

Hence, the model of average total cost per unit time for multi-item joint ordering is defined as below:

$$\min \quad \mathrm{TC}(t, X_1, X_2, \ldots, X_n) = \frac{B + \sum_{i=1}^{n} b_i / X_i}{T} + \sum_{i=1}^{n} \frac{R_i X_i t H_i}{2},$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^{n} R_i X_i t \le V, \\ \min\{X_i, i = 1, 2, \cdots, n\} = 1. \end{cases} \tag{4}$$

In the objective function (4), the first term indicates the total cost of multi-item joint ordering per unit time; the second term represents the inventory cost of various materials.

Formula (4) represents the constraints; the first constraint ensures that the total order quantity at the same time

Table 1: Notations and definitions of parameters.

Notations	Definition
n	The number of types of cruise ship supplies
i	Material i, $i = 1, 2, \ldots, n$
R_i	The demand rate of material i
H_i	Unit time inventory cost per unit of material i in the cruise distribution center
B	Basic ordering cost per order
b_i	Individual ordering cost of material i
V	Inventory capacity of the cruise distribution center
t	Basic order cycle (decision variable)
T_i	The order cycle of material i
X_i	The order cycle of material i is X_i times the basic order cycle (decision variable)
Q_i	The order quantity of material i (decision variable)

does not exceed the inventory capacity of the cruise distribution center; the second constraint indicates that the order cycle of at least one item is equal to the base order cycle of the cruise distribution center.

3.3. Model Solution. Firstly, X_i is regarded as a constant in equation (4), and then we compute the first derivative of equation (4) with respect to t, as shown below:

$$\frac{\mathrm{dTC}(t)}{\mathrm{d}t} = -\frac{B + \sum_{i=1}^{n} b_i/X_i}{t^2} + \sum_{i=1}^{n} \frac{R_i X_i H_i}{2}. \tag{5}$$

As $(\partial^2 \mathrm{TC}(t))/(\partial^2 t) = (B + \sum_{i=1}^{n} b_i/X_i)/(t^3) > 0$, equation (5) can be set to zero, and then the basic order cycle can be expressed as follows:

$$t^*(X_1, X_2, \ldots, X_n) = \sqrt{\frac{2(B + \sum_{i=1}^{n} b_i/X_i)}{\sum_{i=1}^{n} R_i X_i H_i}}. \tag{6}$$

According to the constraints (4),

$$\sum_{i=1}^{n} R_i X_i t \le V. \tag{7}$$

Thus,

$$t \le \frac{V}{\sum_{i=1}^{n} R_i X_i}. \tag{8}$$

Suppose $t > V/\sum_{i=1}^{n} R_i X_i$, then $t^* = V/\sum_{i=1}^{n} R_i X_i$.

Next, we can get the formula simplified by substituting equation (6) into equation (4), which is denoted as

$$\mathrm{TC}(X_1, X_2, \ldots, X_n) = \sqrt{2\left(B + \sum_{i=1}^{n} \frac{b_i}{X_i}\right) \sum_{i=1}^{n} R_i X_i H}. \tag{9}$$

Thus, the problem is to solve the optimal order cycle multiples for the various items, so as to minimize the average total cost of the system. According to equation (9), minimization of $\mathrm{TC}(X_1, X_2, \ldots, X_n)$ is equivalent to minimization of the following equation:

$$F(X_1, X_2, \ldots, X_n) = \left(B + \sum_{i=1}^{n} \frac{b_i}{X_i}\right) \sum_{i=1}^{n} R_i X_i H_i. \tag{10}$$

4. Improved Swarm Intelligence Algorithm

4.1. Fireworks Algorithm. Fireworks algorithm (FWA) is a novel swarm intelligence algorithm inspired by observing fireworks in the sky, which is proposed for global optimization of complex functions, in which two types of explosion (search) processes are employed, and the mechanisms for keeping diversity of sparks are also well designed [27]. The initialization of the fireworks algorithm is a process of generating fireworks randomly. Good explosion is in a small range with plenty of sparks, and the explosion operator is applied to the generated fireworks to generate new sparks. The core of FWA is explosion operator, including explosion intensity, explosion amplitude, and displacement operation.

4.1.1. Number of Sparks. The number of sparks generated by each firework x_i is defined as below:

$$S_i = m * \frac{Y_{\max} - f(x_i) + \varepsilon}{\sum_{i}^{N} (Y_{\max} - f(x_i)) Y_{\max} + \varepsilon}, \tag{11}$$

where m is a parameter controlling the total number of sparks generated by the N fireworks; S_i is the number of sparks generated by firework i; $Y_{\max} = \max(f(x_i))$ $(i = 1, 2, \ldots, N)$ is the maximum (worst) value of the objective function among the N fireworks; $f(x_i)$ is the fitness value of individual x_i; and ε, which denotes the smallest constant in the computer, is utilized to avoid zero-division-error.

4.1.2. Amplitude of Explosion. Amplitude of explosion for each firework x_i is defined as follows.

In contrast to the design of sparks number, the amplitude of a good firework explosion is smaller than that of a bad one. The amplitude of explosion is defined as follows:

$$A_i = \widehat{A} * \frac{f(x_i) - Y_{\min} + \varepsilon}{\sum_{i=1}^{N} (f(x_i) - Y_{\min}) + \varepsilon}, \tag{12}$$

where $\widehat{A}$ denotes the maximum explosion amplitude and $Y_{\min} = \min(f(x_i))(i = 1, 2, \ldots, N)$ is the minimum (best) value of the objective function among the N fireworks.

4.1.3. Generating Sparks. In explosion, sparks may undergo the effects of explosion from random Δx_i^k directions (dimensions); we obtain the number of the affected directions randomly as follows:

$$\Delta x_i^k = x_i^k + \text{rand}(0, A_i), \tag{13}$$

where x_i^k is the dimensionality of location x_i and $\text{rand}(0, A_i)$ is a uniform distribution over $(0, A_i)$.

4.1.4. Spark Gaussian Explosion. Gaussian sparks are generated in a Gaussian explosion process, and Gaussian sparks conducts search in a local space around a firework which can be written as

$$x_i^k = x_i^k G, \tag{14}$$

where G is a random number of Gaussian distributions with mean 1 and variance 1, that is, $G \sim N(1, 1)$.

4.1.5. Selection of Locations. The selection probability of a location x_i is denoted as follows:

$$p(x_i) = \frac{R(x_i)}{\sum_{j \in K} R(x_j)}, \tag{15}$$

where

$$R(x_i) = \sum_{j \in K} d(x_i, x_j) = \sum_{j \in K} \|x_i - x_j\|. \tag{16}$$

In equations (15) and (16), $R(x_i)$ is the sum of the distances between x_i and other individuals; $j \in K$ means that the first location j belongs to set K, which is the set of sparks generated by explosion operator and Gauss mutation; $p(x_i)$ is the probability that each individual chooses to use roulette gamblers; and $d(x_i, x_j)$ is the Euclidean distance between any two individuals x_i and x_j.

4.2. Improved Fireworks Algorithm. The fireworks algorithm with inertia weight (WFWA) is a new algorithm based on the FWA, which adds inertia weight to the spark position update formula as shown in equations (17) and (18) and improves the explosion radius by inertia weight [28]. A fireworks algorithm with nonlinear inertia weight $w(t)$ is introduced to accelerate the convergence speed and improve the performance of the algorithm, in which the inertia weight varies with the optimization ability. The larger the value of inertia weight, the stronger the global optimization ability; when it is small, the algorithm is local optimization. Compared with PSO, GA, FOA, and other swarm intelligence algorithms, WFWA has better convergence, global optimization, and search ability [28–35]. Therefore, WFWA is a good choice to solve the ordering optimization problem of cruise ship supply to the integrated purchase, delivery, and inventory in the supply chain environment.

$$\widehat{x}_j^k = w\widehat{x}_j^k + A_j \cdot \text{rand}(-1, 1)\left(j = 1, 2, \ldots, \widehat{s}_i\right), \tag{17}$$

$$w = \left(\frac{1}{2}\right)^t, \tag{18}$$

where t represents the number of evaluations and w decreases with the number of iterations.

The flowchart of WFWA is shown in Figure 3.

5. Experimental Testing and Analyses

5.1. Experimental Data. We select the data of a cruise company for empirical test and analysis, which wants to purchase a batch of supplies from all over the world, including ten categories with different ordering cycles. The ship supplies require delivery from different suppliers (other warehouses around) to a cruise distribution center, in which they would be assembled and delivered to the cruise ships because separate distribution will inevitably increase additional costs and cause congestion at the loading port. The relevant data of cruise ship supplies purchasing are shown in Table 2, and the basic order cost B is 50 yuan; the storage capacity of cruise distribution center is 600 pieces.

5.2. Simulation Results and Analysis. In this section, in order to verify the effectiveness of the improved swarm intelligence algorithm (WFWA) in solving the procurement optimization problem, the simulation results of WFWA are compared with the fireworks algorithm (FWA), genetic algorithm (GA), and particle swarm optimization (PSO). In the environment of MATLAB, the data in Table 2 are taken as an example for simulation experiment. The number of iterations is 500 times, and each algorithm runs 20 times independently. The parameters of WFWA and FWA are as follows: population size N is 5, constant m is 50, explosion number is 6, explosion radius is 40, explosion limiting factor $a = 0.04$ and $b = 0.8$, Gauss variation spark number is 5, and function variable dimension D is 10. The particle number is 10, and gama parameter is 0.95 for PSO. And the parameters of GA are as follows: the population size is 10, the crossover probability is 0.75, and the mutation probability is 0.75.

The simulation results of the four swarm intelligence algorithms are shown in Figure 4, which indicates that WFWA has the fastest convergence rate and the highest efficiency in the optimum results. Compared with WFWA, FWA and PSO are slower, and GA is the slowest. According to the analysis of simulation results compared with FWA, PSO, and GA, Table 3 lists the testing results for optimal value, worst value, average optimal value, and standard deviation, which evaluate the performance and stability of the algorithms. It can be concluded from Table 3 that the proposed WFWA performs better than FWA, PSO, and GA in optimal solution and average optimal solution and has better robustness. Compared with the other three algorithms, WFWA has better convergence rate and accuracy and can better search for the global optimal solution.

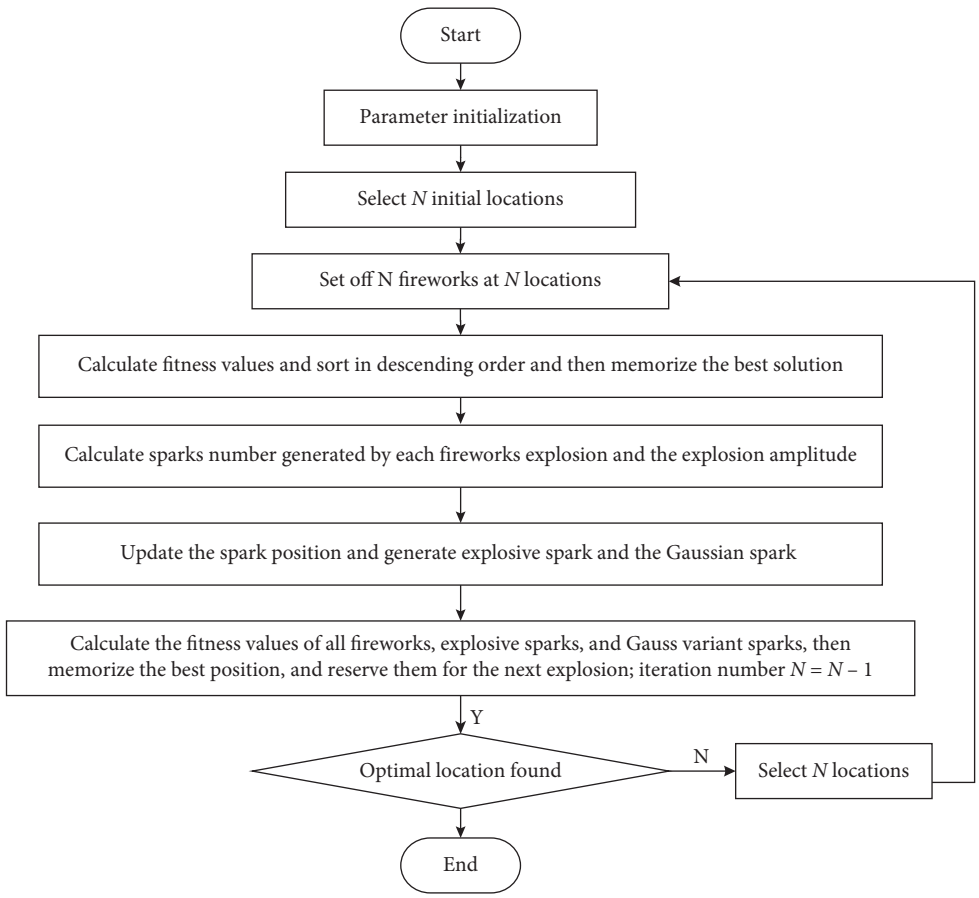


Figure 3: The flowchart of WFWA.

Table 2: Cruise ship supplies purchase list details.

Supplies	Individual ordering cost b_i (yuan/time)	Demand rate R_i (parts/quarters)	Inventory Cost H_i (yuan/piece)
1	80	1700	8
2	100	1200	10
3	50	1500	10
4	150	1000	17
5	200	400	15
6	180	1200	26
7	280	200	5
8	220	500	35
9	250	800	10
10	400	600	14

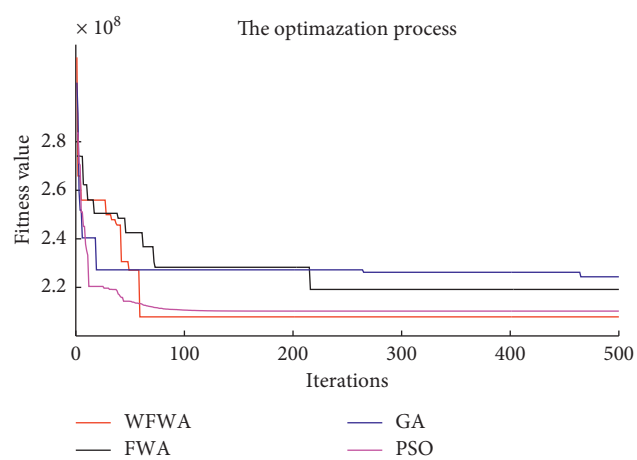


Figure 4: Simulation optimization process of different swarm intelligence algorithms.

WFWA is used to calculate the order cycle of each item of cruise ship supplies; the results are shown in Table 4.

According to the constraint $\sum_{i=1}^{n} R_i X_i t \leq V$, the basic order cycle can be calculated as the follows:

$$t > \frac{V}{\sum_{i=1}^{n} R_i X_i} = 0.0283. \tag{19}$$

As a result, T^* is 0.0283 quarter, namely, 3 days.

It can be seen from Table 4 that after the adjustment of X_i, the order cycle of material 3 is 3 days; the order cycle of materials 1, 4, 6, and 8 is 6 days; the order cycle of material 2 is 9 days; the order cycle of material 9 is 12 days; the order cycle of materials 5 and 7 is 15 days; and the order cycle of material 10 is 24 days.

Therefore, the order quantity of each kind of cruise ship supplies can be calculated according to equation (2), and the results are shown in Table 5.

Finally, the total procurement cost is as follows:

Table 3: The results of simulation experiment.

Algorithms	WFWA	FWA	GA	PSO
Optimal solution	$2.079e+08$	$2.192e+08$	$2.244e+08$	$2.155e+08$
Worst value	$3.148e+08$	$2.741e+08$	$3.044e+08$	$3.232e+08$
Average optimal solution	$2.125e+08$	$2.261e+08$	$2.274e+08$	$2.178e+08$
Standard deviation	$1.372e+07$	$1.105e+07$	$5.498e+06$	$8.951e+06$

Table 4: Calculation result of cruise ship supplies order cycle.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
Calculation	2.2806	3.2968	1.3011	2.4346	4.8441	2.2445	5.2910	2.4576	3.6690	8.4158
Modulation	2	3	1	2	5	2	5	2	4	8

Table 5: Calculation result of cruise material order quantity.

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}
Order quantity	96	102	42	57	57	68	28	28	91	136

$$\mathrm{TC} = \frac{B + \sum_{i=1}^{n} b_i / X_i}{T^*} + \sum_{i=1}^{n} \frac{Q_i H_i}{2} = 28077 \text{ Yuan}. \tag{20}$$

6. Conclusions

With the globalization of cruise ship supplies purchasing, multisourcing brings our attention to multi-item joint ordering in cruise ship supply logistic; this paper considers the optimization objective of inventory collaborative operation in purchase and delivery from the perspective of supply chain and establishes a model of multi-item joint ordering based on a cruise distribution center to minimize the logistics procurement cost for cruise ship supply. And most swarm intelligence algorithms have slow convergence rate and are easily trapped in local optimum; an improved intelligence algorithm WFWA is proposed to solve it. The comparison with GA, PSO, and FWA verifies the effectivity and efficiency of WFWA. Basic order cost and inventory capacity are two key initial values in the model, which are closely related to the order cycle. And as the basic order cost value decreases, the gap between the order cycle of each material and the basic order cycle increases, and the total cost decreases. Meanwhile, the advantage of multi-item joint ordering will be more visible. At the same time, with the decrease of inventory capacity, the basic order cycle will decrease, and the order cycle of each material will also reduce, causing the cost to rise. However, excessive inventory capacity would lead to waste of warehouse resources and increased storage cost. This model considers purchase cost, inventory cost, and delivery cost, which has practical significance for the optimization of cruise ship supply logistics in China. Nevertheless, because of large quantity, high efficiency, and low unit value of the cruise ship supply at the cruise home port, ship suppliers should have sufficient inventory to response to the orders from cruise company, which would cause corresponding inventory costs and other logistics costs for ship suppliers. For the ship suppliers of a commodity supply interruption, the shortage cost can be immeasurable in a certain extent for cruise company. Hence, the optimization problem of interruption risk is also worthy of study in the purchasing process of cruise supply chain.

Data Availability

The data used to support the findings of this study are included in the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (grant no. 51279153).

References

[1] D. Vukonic, D. Bielic, and A. Russo, "Organizational factors in management of "Mega Cruise Ships" from crowd management control aspect," *Scientific Journal of Maritime Research*, vol. 30, no. 1, pp. 58–66, 2016.

[2] J. W. Jones, "Accounting practices in ship chandlery," *Management Accounting*, vol. 55, no. 2, p. 28, 1973.

[3] S. Véronneau and J. Roy, "Global service supply chain: an empirical study of current practices and challenges of a cruise line corporation," *Tourism Management*, vol. 30, no. 1, pp. 129–139, 2009.

[4] G. K. Vaggelas and I. N. Lagoudis, "Analysing the supply chain strategy of the cruise industry-the case of a small cruise company," in *Proceedings of the International Association of Maritime Economists (IAME 2010)*, Lisbon, Portugal, July 2010.

[5] S. Véronneau and Y. Cimon, "Maintaining robust decision capabilities: an integrative human-systems approach," *Decision Support Systems*, vol. 43, no. 1, pp. 127–140, 2007.

[6] P. Lois, J. Wang, A. Wall, and T. Ruxton, "Formal safety assessment of cruise ships," *Tourism Management*, vol. 25, no. 1, pp. 93–109, 2004.

[7] C. X. Fu and J. M. Shou, "The global cruise material supply chain research," *Jiangsu Commercial Forum*, vol. 6, pp. 21–24, 2015.

[8] H. J. Liu, "Discussion on how to formulate the material supply chain management system in cruise industry operation,"

Logistics Engineering and Management, vol. 39, no. 12, pp. 55–58, 2017.

[9] X. Y. Yin, "Experience of international cruise material distribution in home port," *Shipping Management*, vol. 7, pp. 25–28, 2013.

[10] J. Zheng, H. Zhang, L. Yin et al., "A voyage with minimal fuel consumption for cruise ships," *Journal of Cleaner Production*, vol. 215, pp. 144–153, 2019.

[11] J. L. Balintfy, "On a basic class of multi-item inventory problems," *Management Science*, vol. 10, no. 2, pp. 287–297, 1964.

[12] S. K. Goyal, "Determination of economic packaging frequency for items jointly replenished," *Management Science*, vol. 20, no. 2, pp. 232–235, 1973.

[13] E. A. Sivler, "A simple method of determining order quantities in joint replenishments under deterministic demand," *Management Science*, vol. 22, no. 12, pp. 1351–1361, 1976.

[14] H. Pirkul and O. A. Aras, "Capacitated multiple item ordering problem with quantity discounts," *IIE Transactions*, vol. 17, no. 3, pp. 206–211, 1985.

[15] C. J. Xu and E. Xu, "A multi-item ordering tactics at regular intervals in reverse logistics," *Logistics Technology*, vol. 5, pp. 28-29, 2006.

[16] L. Zhu, J. T. Zhang, and J. X. You, "Study on supply chain aggregation effects based on variety of products or customers ordering strategy," *Journal of Tongji University: Natural Science*, vol. 37, no. 1, pp. 134–138, 2009.

[17] X. W. Wen and Q. L. Da, "Ordering policy and empirical study of multi-product in a supply chain," *Journal of Industrial Engineering and Engineering Management*, vol. 21, no. 1, pp. 56–60, 2007.

[18] J. L. Zhang, L. Wang, T. Chen, and Y. R. Zeng, "Coordinated replenishment problem of rarely used spare parts based on continue production," *Chinese Journal of Management Science*, vol. 12, no. 5, pp. 58–62, 2004.

[19] J. R. Zhang, Y. L. Yu, Y. L. Zhang, and C. C. Qu, "Joint order model of multi-items maintenance spare parts based on periodic policy," *Fire Control & Command Control*, vol. 37, no. 9, pp. 112–116, 2012.

[20] Z. M. Guan and H. Wu, "Optimized model for order quantity allocation of multi-product purchase with fuzzy demand," *Journal of Northeastern University: Natural Science*, vol. 33, no. 8, pp. 1198–1201, 2012.

[21] N. K. Verma, A. Chakraborty, and A. K. Chatterjee, "Joint replenishment of multi retailer with variable replenishment cycle under VMI," *European Journal of Operational Research*, vol. 233, no. 3, pp. 787–789, 2014.

[22] I. K. Moon, S. K. Goyal, and B. C. Cha, "The joint replenishment problem involving multiple suppliers offering quantity discounts," *International Journal of Systems Science*, vol. 39, no. 6, pp. 629–637, 2008.

[23] L. Cui, L. Wang, J. Deng, and J. Zhang, "Intelligent algorithms for a new joint replenishment and synthetical delivery problem in a warehouse centralized supply chain," *Knowledge-Based Systems*, vol. 90, pp. 185–198, 2015.

[24] Y.-C. Tsao and W.-G. Teng, "Heuristics for the joint multi-item replenishment problem under trade credits," *IMA Journal of Management Mathematics*, vol. 24, no. 1, pp. 63–77, 2013.

[25] P. Ongkunaruk, M. I. M. Wahab, and Y. Chen, "A genetic algorithm for a joint replenishment problem with resource and shipment constraints and defective items," *International Journal of Production Economics*, vol. 175, pp. 142–152, 2016.

[26] B. Marcello, C. Davide, and G. Mose, "An extension of the stochastic Joint-Replenishment Problem under the class of cyclic policies," *Operations Research Letters*, vol. 44, no. 2, pp. 278–284, 2016.

[27] Y. Tan, *Fireworks Algorithm: A Novel Swarm Intelligence Optimization Method*, Springer, Berlin, Germany, 2015.

[28] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," *Lecture Notes in Computer Science*, vol. 6145, pp. 355–364, 2010.

[29] Y. F. Shang, X. Y. Liu, and M. N. Jia, "Fireworks algorithm with inertia weight," *Journal of Inner Mongolia University of Technology*, vol. 35, no. 3, pp. 168–177, 2016.

[30] W. Guo, J. Li, G. Chen, Y. Niu, and C. Chen, "A PSO-optimized real-time fault-tolerant task allocation algorithm in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3236–3249, 2015.

[31] W. Guo, G. Liu, G. Chen, and S. Peng, "A hybrid multi-objective PSO algorithm with local search strategy for VLSI partitioning," *Frontiers of Computer Science*, vol. 8, no. 2, pp. 203–216, 2014.

[32] G. Liu, W. Guo, Y. Niu, G. Chen, and X. Huang, "A PSO-based timing-driven Octilinear Steiner tree algorithm for VLSI routing considering bend reduction," *Soft Computing*, vol. 19, no. 5, pp. 1153–1169, 2015.

[33] L. Guo, M. Li, and D. Xu, "Efficient approximation algorithms for maximum coverage with group budget constraints," *Theoretical Computer Science*, vol. 788, pp. 53–65, 2019.

[34] G. Liu, W. Guo, R. Li, Y. Niu, and G. Chen, "XGRouter: high-quality global router in X-architecture with particle swarm optimization," *Frontiers of Computer Science*, vol. 9, no. 4, pp. 576–594, 2015.

[35] G. G. Liu, X. Huang, W. Z. Guo, Y. Z. Niu, and G. L. Chen, "Multilayer obstacle-avoiding X-architecture steiner minimal tree construction based on particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 989–1002, 2015.