Hindawi
Complexity
Volume 2021, Article ID 6061939, 10 pages
https://doi.org/10.1155/2021/6061939

Research Article

Vehicle Type Recognition Algorithm Based on Improved Network in Network

Erxi Zhu,[1,2] **Min Xu**,[3,4] **and De Chang Pi**[2]

[1]*College of Internet of Things Engineering, Jiangsu Vocational College of Information Technology, Wuxi 214153, China*
[2]*College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 211106, China*
[3]*College of Electronic and Information Engineering, Jiangsu Vocational College of Information Technology, Wuxi 214153, China*
[4]*Jiangsu Key Laboratory of ASIC Design, Wuxi 214153, China*

Correspondence should be addressed to Erxi Zhu; erxi666@163.com

Received 7 July 2020; Revised 19 November 2020; Accepted 22 December 2020; Published 5 January 2021

Academic Editor: Jia Wu

Vehicle type recognition algorithms are broadly used in intelligent transportation, but the accuracy of the algorithms cannot meet the requirements of production application. For the high efficiency of the multilayer perceptive layer of Network in Network (NIN), the nonlinear features of local receptive field images can be extracted. Global average pooling (GAP) can avoid the network from overfitting, and small convolution kernel can decrease the dimensionality of the feature map, as well as downregulate the number of model training parameters. On that basis, the residual error is adopted to build a novel NIN model by altering the size and layout of the original convolution kernel of NIN. The feasibility of the algorithm is verified based on the Stanford Cars dataset. By properly setting weights and learning rates, the accuracy of the NIN model for vehicle type recognition reaches 97.2%.

1. Introduction

Intelligent transportation [1] refers to a research hotspot in existing society, and vehicle type recognition [2] underpins and critically impacts intelligent transportation studies. The existing algorithms of vehicle type recognition are primarily classified as manual feature descriptions, 3D model, and artificial intelligence algorithms. At the early phase, the manual feature descriptions (e.g., SIFT [3] and HOG [4]) are adopted to extract vehicle features; subsequently, the algorithms (e.g., SVM and decision tree) are combined for classification. Since feature extraction and data reconstruction are difficult to achieve, Hsieh et al. [5] employed HOG and symmetric SURF descriptor to extract the vehicle features of mesh generation. Besides, Liao et al. [6] conducted the appearance and semantic segmentation of vehicle parts to recognize vehicle types. Moreover, Biglari et al. [7] exploited the overall appearance of the vehicles and the feature differences of various components to train the SVM classifier. The mentioned algorithms are easy to affect by environmental factors (e.g., light and background), so their recognition accuracy is relatively low. As impacted by the random variation in the shooting angle of vehicle images, the 3D model-based vehicle type recognition method was developed at the right moment. The 3D model can reflect spatial relationships between local features and the whole vehicle. Existing studies [8, 9] effectively performed the 3D modeling and feature extraction of vehicles. Artificial intelligence introduced a novel impetus into vehicle type recognition, and the features of the vehicle can be automatically extracted. Dong et al. [10] adopted the sparse Laplace filter and a semisupervised convolution neural network to extract vehicle features and classify vehicles. Studies [11–14] employed different methods or optimized the existing neural network to conduct the vehicle type recognition, and its effect was significantly improved; however, for the similar vehicle recognition exhibiting a remarkably small feature gap (e.g., Volkswagen's front face is nearly identical), the room for improvement of classification accuracy is limited.

In view of the low accuracy of vehicle type recognition, we propose an improved NIN for vehicle type recognition and get

high recognition accuracy. In fact, the breakthrough point of vehicle type recognition refers to the efficient extraction of nonlinear features of vehicles. NIN [15] exhibits a complex multilayer perceptron (MLPConv) with a micronetwork structure and is capable of efficiently and automatically extracting local nonlinear features of images. The present study fully exploits the following features of the NIN model and uses its 1×1 convolution kernel to conduct the dimensionality reduction of the feature map and downregulate the number of network parameters. The global average pooling layer (GAP) is adopted to effectively combine the features and prevent the whole network from falling into the overfitting state. The improvement measures are as follows: the original large convolution kernel of NIN is changed into a small convolution kernel, which increases the depth of convolution neural network and improves the performance of the network. In order to avoid the gradient loss problem caused by the increase of depth, residual measures are arranged on the structure to solve the network degradation problem. The improved NIN has high classification effects, and its classification accuracy is better than VGG and GoogLeNet in vehicle type recognition. By the verification based on the Stanford Cars dataset and the reasonable weight and learning rate setting, the vehicle type recognition accuracy of the improved NIN reaches over 97.2%.

2. Related Works

The 1×1 small convolution kernel, GAP, micronetwork structure, and other measures proposed by NIN underpin the follow-up deep convolutional neural network (CNN). CNN [16] automatically extracts image features; thus, the complex feature extraction and data reconstruction process of conventional recognition algorithms can be avoided. AlexNet [17], VGGNet [18–21], GoogLeNet [22, 23], ResNet [24–27], and other networks can be adopted for vehicle type recognition, whereas for the limitations of sample quality and quantity as well as the defects of network feature extraction and classification performance, vehicle recognition exhibits relatively low accuracy.

Most networks are only capable of extracting linear features on the images, landing the classification algorithm in confusion since the linear features are basically consistent (Figure 1(a) and (b)). For classification, only the overall information built by linear features can be classified (Figure 1(c)).

In Figure 1, the linear features denoted by (a) and (b) are consistent, which are both a line segment and a part of an object without any difference. However, given the overall information, the information represented by (c) is completely inconsistent. Thus, a question is raised of how to extract this nonlinear feature effectively. This question is determined by the micronetwork [28, 29] structure embedded in NIN, i.e., a full connection layer consisting of two layers of convolution. In the neural network, two-layer fully connected hidden neurons are capable of approximating arbitrary curves.

2.1. "Micronetwork" Structure. In 2013, the proposal of NIN modified the original idea of network structure, and the multilayer perceptron was built by replacing the conventional linear perceptron with the embedded "micronetwork"; as a result, the efficiency of nonlinear feature extraction of local sensing field of images was significantly enhanced.

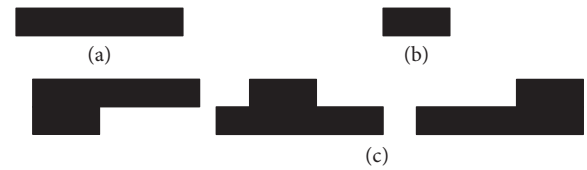

Figure 1: Schematic diagram of linear features and linear feature combinations.

In NIN, "micronetwork" refers to a general nonlinear function approximator. The difference between MLPConv of NIN and linear perceptron of CNN is the method of image feature extraction. MLPConv consists of several fully connected nonlinear activation functions, shared by all local receptive fields. Moreover, by sliding on the input, the feature map is generated and then outputted to the next layer. MLPConv can combine different feature maps, so the network can extract complex and useful nonlinear image features. Furthermore, the overall structure of NIN can be superposed by multiple MLPConv.

There are two reasons why NIN selects multilayer perceptron: (1) MLPConv fits the structure of the convolutional neural network and (2) MLPConv can act as a deep model, complying with the spirit of feature reuse [22]. The feature map of MLPConv is calculated:

$$\begin{aligned} f^1_{i,j,k_1} &= \max\left(\omega^{1\ T}_{k_1} x_{i,j} + b_{k_1}, 0\right), \\ &\vdots \\ f^n_{i,j,k_n} &= \max\left(\omega^{n\ T}_{k_n} f^{n-1}_{i,j} + b_{k_n}, 0\right), \end{aligned} \tag{1}$$

where n denotes the number of layers of the multilayer perceptron; (i, j) represents the pixel index in the feature map; $x_{i,j}$ indicates the input block centred on the position (i, j); k is the channel index of the feature map; and b_{k_1} is the bias. ReLU acts as the activation function in MLPConv.

2.2. Global Average Pooling Layer. In the classification, GAP [30, 31] remedies the defect of the fully connected layer. At the early phase, the feature map of the final convolutional layer is vectorized and passed into the fully connected layer; subsequently, it is inputted to the SoftMax layer [32–34]. Since the fully connected layer is easy to overfit, the whole network exhibits a reduced generalization ability, and the subsequent network conducts a dropout [24] operation on the fully connected layer, thereby preventing overfitting significantly. However, GAP is adopted by NIN to set the last MLPConv feature map to pertain to the corresponding classification category, which can more effectively fit the convolution structure. There are no parameters to be optimized in the operation, thereby avoiding overfitting. The regularization effect of GAP is more significant than dropout.

2.3. 1×1 *Convolution Kernel.* The 1×1 convolution was initially proposed by NIN to make the network exhibit significantly high network performance. By 1×1 convolution computation, MLPConv reduces the dimension of the channel parameter pool of convolutional kernel, as well as downregulating the number of parameters. The main functions of 1×1 convolution are as follows:

(1) Dimensionality reduction: for instance, if an 500×500 image with a depth of 100 is generated with 1×1 convolution on 20 filters, the size of the result is $500 \times 500 \times 20$.

(2) The nonlinear expression ability is enhanced. After the convolutional layer passes through the excitation layer, the 1×1 convolution introduces nonlinear excitation to the learning representation of the previous layer to enhance the expression ability of the network.

(3) The model depth is increased. Accordingly, the number of the network model parameters can be reduced, the depth of the network layer can increase, and the representational capacity of the model can be enhanced to some extent.

Figure 2 illustrates the NIN structure of 4 MLPConv and 1 GAP. Subsampling layers can be added between MLPConv layers, and the number of layers of the "micronetwork" can be altered for specific tasks. First, taking the first MLPConv as an example, the input image is $224 \times 224 \times 3$, 224 represents the pixel of the input image, and 3 denotes the channel of the image. Later, the convolution filter is adopted to slide on the input image and calculate the inner product. The size of the convolution filter adopts $11 \times 11 \times 3$, i.e., the length and width are both 11, and the depth is 3. In the first layer of MLPConv, 96 convolution filters are adopted. The embedded "micronetwork" refers to a fully connected neural network with a two-layer convolutional kernel, performing nonlinear feature extraction. The number of neurons in each layer reaches 96. Besides, Figure 2 presents one of the models compared in subsequent experiments, and the specific setting of parameters is presented in the figure.

In the present study, the nonlinear feature extraction capacity of NIN is exploited to extract the features of vehicles in the image (e.g., texture and topology structure) to enhance the efficiency of the vehicle type recognition. On that basis, by increasing the size, quantity, and layout of the convolutional kernel in NIN, as well as the network performance and convergence speed, the training of NIN for vehicle sample data is conducted efficiently, and the vehicle recognition accuracy is enhanced. Subsequently, the residual thought is adopted to solve gradient dissipation that is attributed to the rising number of network layers.

3. Optimized NIN

At present, network performance can be enhanced primarily by two measures. One is to increase the width or depth of the network. For instance, VGG enhances network performance by increasing network depth. The other refers to optimizing the network input sample data (e.g., increasing the sample number, strengthening the texture of the sample, or transforming the shape of the sample image (inversion and distortion) to enhance the network performance). For the deepened or widened network, its defects gradually appear, the gradient disappears, the number of parameters is huge, and the extracted features tend to be invalid in the network transmission. In the present study, NIN is optimized by the following two means.

3.1. Use of Small Convolution Kernel. The small convolution kernel increases the network depth and improves the network performance, as well as significantly downregulates the number of network parameters. In numerous networks, the convolution kernel with a size of 3×3 and 5×5 has been extensively used, and 3×3 refers to the smallest size that can capture 8 neighbourhood information of pixels.

The small convolution kernels are stacked to replace the large convolution kernels, and the size of the receptive field remains unchanged. Multiple 3×3 convolution kernels exhibit more nonlinearities (more layers of nonlinear functions) than the convolution layer of a large convolution kernel. Moreover, multiple 3×3 convolutional layers have fewer parameters than a large convolution kernel. If the input and output feature maps of the convolutional layer are assumed to have an identical size to C, the number of parameters of the three convolutional layers is $3 \times (3 \times 3 \times C \times C) = 27C^2$. The parameter of one 7×7 convolutional layer is $49C^2$. Thus, the small convolution kernel significantly reduces the number of network parameters.

At the beginning of AlexNet and NIN training, a large convolution kernel is employed for calculation, and the classification accuracy is not significantly enhanced. Even though NIN employs a micronetwork as a local nonlinear feature collector, it only increases the convergence speed of the model. On the whole, the convolution kernel of VGG uses 3×3 convolution kernel, and GoogLeNet contains 3×3, 5×5, and 1×1; the classification effect of VGG and GoogLeNet models is larger than that of the former two. Indeed, this is also attributed to the deepening of the number of network layers. The function of 1×1 convolution kernel suggested that it exhibits the function of raising and reducing dimension and can downregulate the number of network parameters in Section 2.

An experiment is performed to verify the influence of small convolution on the model classification. MINST dataset is employed in the experiment, and the network structure is adopted (Figure 3). The experiment is split into two groups to verify the effect of 7×7, 5×5, and 3×3 convolution kernels on the network performance, respectively. The statistics is summarized to the iteration times under the accuracy of the four models reaching over 0.6, 0.7, 0.8, and 0.9 initially, as well as the iteration times in the presence of maximum accuracy as well as the maximum accuracy and time consumed initially. Each model experiment is repeated 50 times, and the average number of statistical iterations is listed in Table 1.

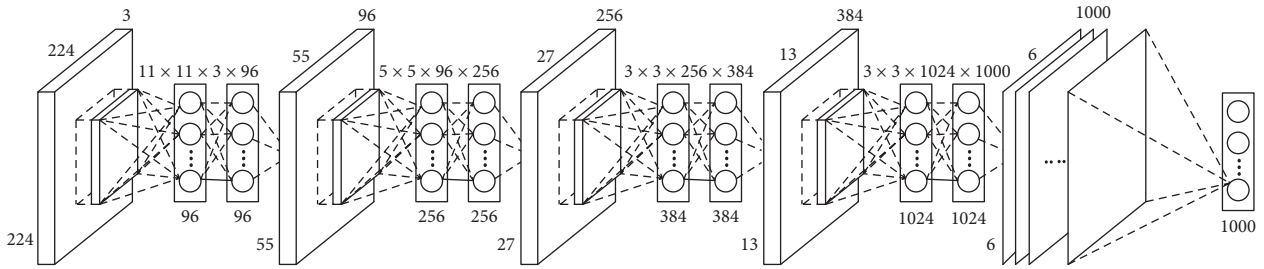

Figure 2: Structure and specific parameter settings of NIN.

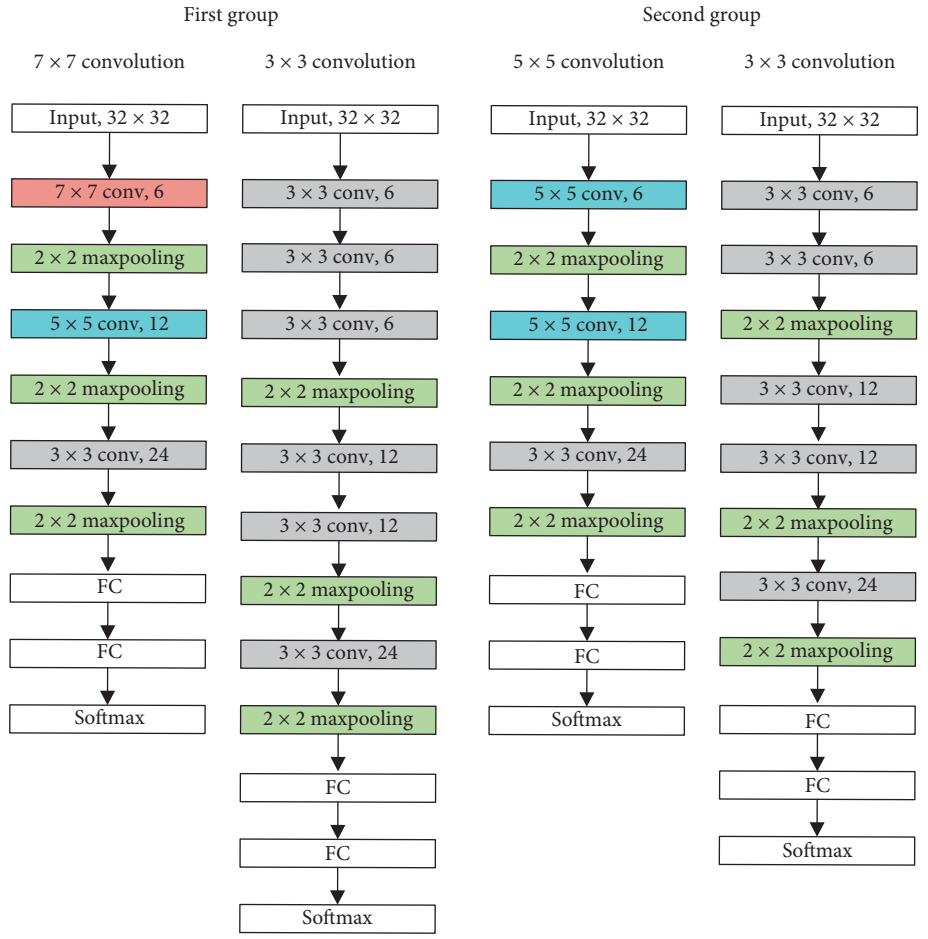

Figure 3: Network structure of small convolutional kernel experiment.

Table 2 presents that the small convolution kernel enhances the extraction performance of local receptive field features of the network and increases the classification accuracy of the model. Three 3×3 convolution kernels are equivalent to a 7×7 convolution kernel, and two 3×3 convolution kernels are equated with a 5×5 convolution kernel. Under the receptive field of the identical convolution kernel, it is easy to find by comparison that the recognition efficiency of the convolution kernel falls to the maximum. In all effective intervals, the average number of experimental iterations of 5×5 convolution kernel is smaller than that of 3×3 and 7×7 convolution kernels. 3×3 convolution kernel exhibits the highest accuracy, whereas the accuracy of 5×5 convolution kernel is relatively low; however, the convolution kernel exhibits significantly low accuracy. Accordingly, in general, 3×3 convolution kernel has the maximum recognition efficiency and the fastest rise in accuracy; that is, 3×3 convolution kernel exhibits a better performance to extract local features of images.

To obtain the vehicle type recognition accuracy, the NIN structure is optimized. The size, quantity, and layout of the convolution kernel of the NIN structure in Section 2 are tuned in accordance with the advantages of the small convolution kernel to extract local features of the image and downregulate

Table 1: Experimental data of small convolution kernel.

Convolution form	Exceeding the average number of iterations of accuracy for the first time					Maximum accuracy rate	Time (h)
	0.6	0.7	0.8	0.9	Maximum		
(1) 7*7	100.9	157.1	230.6	320.1	832.2	0.845	1.261
(1) 3*3	88.4	148	207.6	293.8	799.3	0.965	1.469
(2) 5*5	75.3	114.5	175.2	252.6	821.6	0.903	1.328
(2) 3*3	74.4	110.0	172.6	249.8	819.3	0.914	1.409

Table 2: Comparison of experimental results on the Stanford Cars dataset.

Network name	1		2		3		4		5	
	Accuracy rate (%)	Iterations	Accuracy rate (%)	Iterations	Accuracy rate (%)	Iterations	Accuracy rate (%)	Iterations	Accuracy rate (%)	Iterations
NIN	80.2	5066	88.3	5628	90.4	5822	91.6	5923	91.2	6022
VGG19	83.1	5732	89.1	5913	91.5	6134	92.4	6417	92.7	6982
GoogLeNet	85.1	5522	90.3	5817	92.6	6025	93.0	6120	94.3	6216
New NIN	84.2	4909	90.5	5423	95.5	5781	96.2	5883	97.2	5989

the number of computational parameters of the network. Figure 4 suggests that the 11×11 convolution kernel of the first layer is converted into $4_{3\times3}$ convolution kernels.

3.2. Use of Residual Blocks. Since AlexNet, the depth of the most advanced CNN architecture has been increasing, whereas the depth of the network cannot increase by simply stacking layers. The mentioned finding is because the gradient backpropagates to the previous layer, and repeated multiplication may make the gradient infinitesimal and the gradient disappear; the deep network is difficult to train, and the network performance tends to be saturated, or even drops rapidly. To address this problem, He Kaiming et al. proposed the residual network ResNet; in 2015, the proposed network won the first prize in the challenge competition of ImageNet image recognition and has deeply inspired the design of the later deep neural network.

He Kaiming considered that the training errors produced by stacking identity maps on the deep network should not be higher than those attributed to shallow networks. According to Figure 5, the residual block can achieve the mentioned condition, and the input can be spread by cross-layer data line forward faster. In fact, ResNet is not the first model exploiting fast connection. Highway networks [35] and long and short-term memory network [36] units employ different gate structures to conduct fast connection.

ResNet (Figure 6) continues to use the design of all 3×3 convolution layer of VGG. First, there are two 3×3 convolutional layers with an identical number of output channels in the residual block. Each convolutional layer is followed by a batch normalization layer and ReLU activation function. Subsequently, the input is directly introduced to the front of the final ReLU activation function by skipping the two convolutional operations. In the mentioned design, the output and input of the two convolutional layers should exhibit the identical shape, and then they should be added. To alter the number of channels, an additional 1×1 convolutional layer should be introduced to transform the input into the required shape, and then an addition operation is required.

As impacted by small convolution kernel and residual concept, the NIN is further optimized, and the convolution kernel in NIN is replaced by 3×3 convolution kernel to conduct the rapid convergence and training of the network. The residual measurement is performed to build data lines between the front and back layers of the network, so the feature map can be efficiently transmitted to the front convolutional layer, thereby eliminating the effect of gradient accumulation and decreasing and avoiding gradient disappearance. Given the setting requirements of ResNet, the optimized NIN structure is illustrated in Figure 6.

4. Implementation of Optimized NIN

The optimized NIN uses 3×3 convolution kernel and 1×1 convolution kernel [37, 38]. 3×3 convolution kernel is used to increase network depth and improve network performance. 1×1 convolution kernel is used to enhance the extraction ability of nonlinear features of the network. In the optimized NIN structure, GAP is used as a classifier instead of full connection layer and to improve the generalization ability of the network and avoid overfitting of the network. In order to avoid the loss of gradient caused by the increase of network depth, residual measures are arranged between consecutive multiple 3×3 convolution layers on the optimized NIN to avoid network degradation. The partial source code of optimized NIN is as follows: (Algorithm 1)

5. Results and Discussion

The results and discussion may be presented separately, or in one combined section, and may optionally be divided into headed sections. The representative Stanford Cars dataset is adopted in the experiment. The scene with the images located varies with different postures [39] and unfixed resolutions. Accordingly, the vehicle type recognition of this dataset is more challenging. The Stanford Cars dataset

```
Input:
input_shape: Input shape of network, default as (224,224,3)
nclass: Numbers of class (output shape of network), default as 1000
Output: Optimized NIN model
The optimized NIN model is established according to the following steps:
Step 1: Build two residual blocks including 384 convolution kernels
            Build two 1 × 1 convolution layers
            Build Max pool layer
Step 2: Build two residual blocks including 384 convolution kernels
            Build two 1 × 1 convolution layers
            Build Max pool layer
Step 3: Build residual block including 384 convolution kernels
            Build two 1 × 1 convolution layers
            Build Max pool layer
Step 4: Build residual block including 1024 convolution kernels
            Build two 1 × 1 convolution layers
Step 5: Build GAP layers
            return model
```

Algorithm 1: Partial source code of optimized NIN.

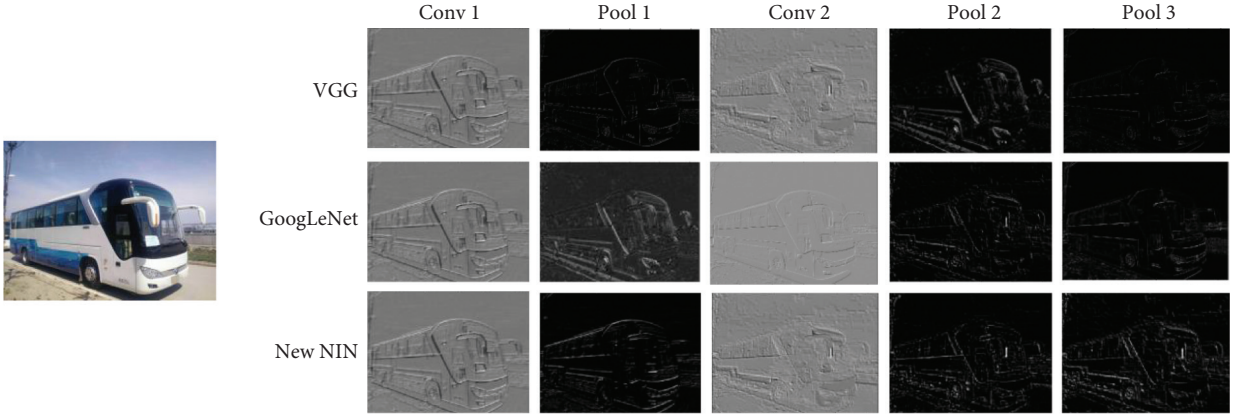

Figure 4: Visualization of activation values in the middle layer of three types of convolutional neural network models.

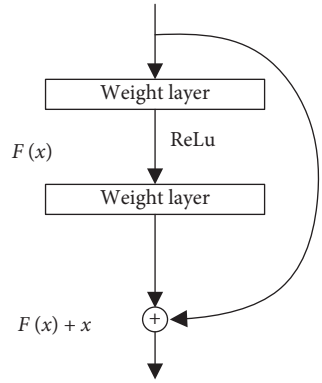

Figure 5: Schematic diagram of the residual block.

consists of 196 vehicle types, containing 16,185 images overall. The dataset labels consist of the vehicle types and the location of the vehicles in the image. The hardware environment of the experiment is presented: CPU type is Xeon W; memory type is DDR4 128GB; graphics card is NVIDIA RTX 2080Ti, and video memory size is 11GB. All the experimental networks are achieved by GPU built by Anaconda 3 + Tensorflow 2.0 + Spyder + Python 3.7 in Windows 10.

To determine the performance of optimized NIN on vehicle type feature extraction, VGG19 (layer 19), GoogLeNet Inception V1 (layer 22), NIN (layer 12), and

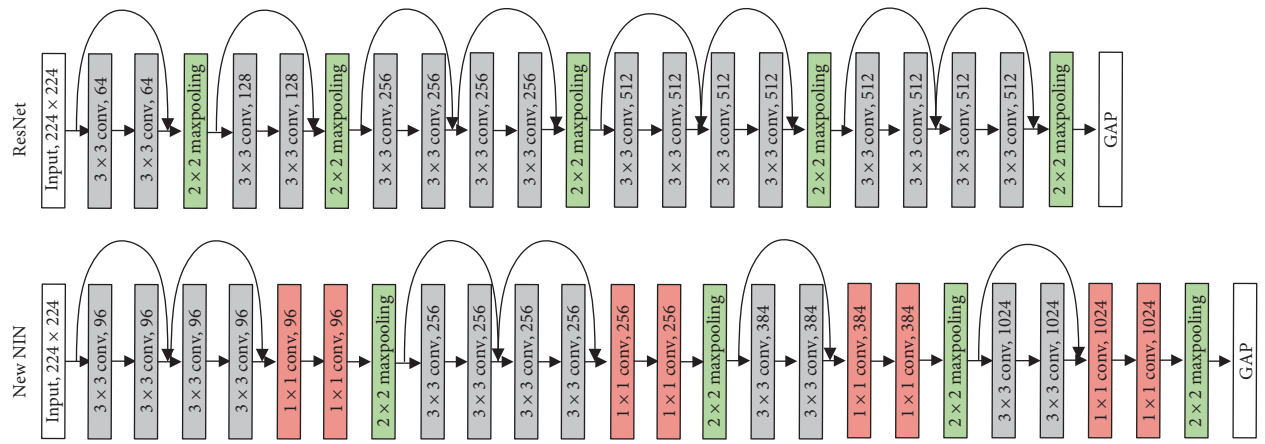

Figure 6: ResNet network structure and optimized NIN structure.

optimized NIN (layer 20) act as the comparison network models. GAP + SoftMax is employed for all the mentioned network model classifiers, and all network training employs data input dimensions. The preprocessing of the dataset, the splitting of the training set, and the verification set comply with literature [40]: the image size of the dataset is normalized to 256×256, 4 corners and the centre part are cut to generate 5 images with a size of 224×224, and the mirror operation is performed to generate 10 training images on the whole, from which the mean value of the training set image is subtracted to obtain the training input data. In the present study, appropriate weights and learning rates are manually set to achieve initialization. The training process starts from the initial weight and learning rate and continues till the accuracy of the training set stops enhancing, and then the learning rate reduces to one-tenth of the original. This process is repeated five times. The weight of the model is updated with the stochastic gradient descent method, and the initial learning rate is 0.01.

5.1. Vehicle Type Recognition Performance. After repeated training of several models, the classification accuracy rate and the number of iterations reached initially are determined from the Stanford Cars sample data, as listed in Table 2.

The optimized NIN has the original MLPConv of NIN. The nonlinear features of the image can be approximated through "micronetwork" structure, so the optimized NIN has fast convergence. By replacing the large convolution kernel of the original NIN with the small convolution kernel, the optimized NIN has deeper layers than the original NIN. The computational effect of multiple 3×3 convolution kernels is equivalent to that of a 5×5 convolution kernel. Using this conversion, all the large convolution kernels of the original NIN are replaced by 3×3 small convolution kernels, which increases the convolution layers of the NIN and enhances the network performance. The residuals are deployed on the NIN structure to avoid the loss of gradient and restrain the degradation of network performance. It can be found from Table 2 that the number of iterations of NIN in each iteration process is less than that of VGG and

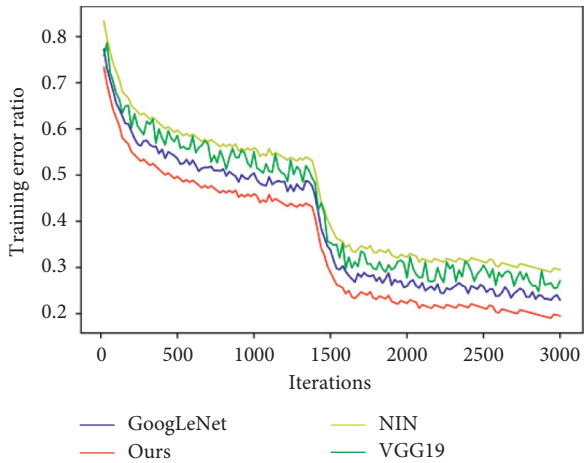

Figure 7: Training error curve.

GoogLeNet, which indicates that the convergence speed of NIN is quicker than that of VGG and GoogLeNet. However, at the end of the experiment, the recognition accuracy of NIN did not exceed that of VGG and GoogLeNet, even if the NIN trained too many iterations. However, the optimized NIN keeps good convergence because the "micronetwork" structure can extract the nonlinear features of the automobile image. In addition, the optimized NIN solves the problem of gradient weakening in the calculation process by the residual layout and strengthens the feature map for subsequent calculation. Therefore, the optimized NIN model outperforms VGG and GoogLeNet in accuracy and convergence speed, and the final vehicle type recognition accuracy reaches 97.2%.

5.2. Vehicle Feature Extraction Capability. VGG19 and GoogLeNet only consist of linear perception layer and only extract linear features [41–45] of vehicles, while NIN and optimized NIN contain multilayer perception layer, which can capture nonlinear features of vehicles. Figure 4 draws the comparison of feature maps of feature extraction of vehicle images after training of several network models.

In Figure 4, Column Conv1 presents the effect of feature map extraction of the three networks after the first convolution kernel operation, column Pool1 refers to the effect of the first pooling layer processing, and column Conv2 represents the sixth-layer convolution calculation results of VGG19 and the third inception structure processing result of GoogLeNet, as well as the second MLPConv processing result of NIN. As revealed from the figure, the ability of the optimized NIN model to extract feature map reaches over those of VGG and GoogLeNet.

5.3. Convergence Effect of Optimized NIN. The experimental data of NIN, VGG19, GoogLeNet, and the optimized NIN in the first 3000 iterations of the third experiment are intercepted, and the training error curves of the sample data of the four networks are plotted (Figure 7).

Figure 7 suggests that the recognition training error of the optimized NIN in the training process is significantly lower than that of the other three networks. In the vicinity of 1300 iterations, the training error of the optimized NIN model did not continue to decrease. We reduce the learning rate of the models participating in the comparison to one-tenth of the original. Each model continued to learn according to the new learning rate, and the training error had a cliff drop in this case, which improves the training speed. In the 3000th iteration, it drops to 19.6%, while the error rate of NIN, VGG19, and GoogLeNet reduces to 31.2%, 28.9%, and 24.6%, respectively. This also indicates that the optimized NIN exhibits good convergence and accelerates the training speed of vehicle license plate recognition.

6. Conclusions

In the present study, the structure and vital components of NIN are analysed, and it is verified that the NIN embedded micronetwork can efficiently extract the nonlinear features of vehicle images, and GAP avoids the overfitting of models and can regularize operation; besides, 1×1 small convolution conducts the dimensionality reduction of feature maps, downregulating the number of model parameters. Based on the NIN, a novel vehicle type recognition algorithm is built by changing the size and layout of the convolution kernel and using residual thought of NIN. Subsequently, it is verified in the Stanford Cars dataset, and the result reveals that the algorithm exhibits a better vehicle type recognition performance and higher recognition accuracy that reaches 97.2%. However, the optimized NIN also has shortcomings. First, in the same local receptive field, the large convolution kernel can be replaced by the small convolution kernel. Although the small convolution kernel operation reduces the number of variables compared with the large convolution kernel operation, the training time is greatly improved, and the efficiency is reduced. Second, the strategy of optimizing NIN is to deepen the network level. To some extent, the application of residual can solve the problem of gradient vanishing and restrain the degradation of network performance. Whether this network performance improvement method can support the further increase of network depth remains to be studied, which also points out the direction for our future work.

Data Availability

The authors used the vehicle dataset provided by Stanford University to verify the improved model. The Cars dataset contains 16,185 images of 196 classes of cars. The data are split into 8,144 training images and 8,041 testing images, where each class has been split roughly in a 50-50 split. Classes are typically at the level of make, model, and year, for example, 2012 Tesla Model S or 2012 BMW M3 coupe; visit http://ai.stanford.edu/~jkrause/cars/car_dataset.html.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by the "Geometry Problem Geometry" project (the National Natural Science Foundation of China (NSFC), 61073086). Some of the authors of this publication are also working on these related projects: (1) Higher Vocational Education Teaching Fusion Production Integration Platform Construction Projects of Jiangsu Province under Grant no. 2019(26), (2) Natural Science Fund of Jiangsu Province under Grant no. BK20131097, (3) "Qin Lan Project" Teaching Team in Colleges and Universities of Jiangsu Province under Grant no. 2017(15), and (4) High Level of Jiangsu Province Key Construction Project funding under Grant no. 2017(17).

References

[1] M. Barth and J. J. Sanchez-Medina, "Guest editorial special issue: the 21st IEEE international conference on intelligent transportation systems (ITSC 2018)," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3929-3930, 2020.

[2] Y. Xiang, Y. Fu, and H. Huang, "Global topology constraint network for fine-grained vehicle recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2918–2929, 2020.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 886–893, IEEE, San Diego, CA, USA, June 2005.

[5] J.-W. Hsieh, L.-C. Chen, and D.-Y. Chen, "Symmetrical surf and its applications to vehicle detection and vehicle make and model recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 6–20, 2014.

[6] L. Liao, R. Hu, J. Xiao, Q. Wang, J. Xiao, and J. Chen, "Exploiting effects of parts in fine-grained categorization of vehicles," in *Proceedings of the IEEE International Conference on Image Processing*, pp. 745–749, IEEE, Quebec City, Canada, September 2015.

[7] M. Biglari, A. Soleimani, and H. Hassanpour, "Part-based recognition of vehicle make and model," *IET Image Processing*, vol. 11, no. 7, pp. 483–491, 2017.

[8] Y. L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis, "Jointly optimizing 3D model fitting and fine-grained classification," in *Proceedings of the European Conference on Computer Vision*, pp. 466–480, Springer, Zurich, Switzerland, September 2014.

[9] J. Krause, M. Stark, J. Deng, and F.-F. Li, "3D object representations for fine-grained categorization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 554–561, IEEE, Sydney, NSW, Australia, December 2013.

[10] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247–2256, 2015.

[11] J. Sochor, A. Herout, and J. Havel, "Boxcars: 3D boxes as cnn input for improved fine-grained vehicle recognition,," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3006–3015, IEEE, Las Vegas, NV, USA, June 2016.

[12] Y. B. Gao and H. J. Lee, "Local tiled deep networks for recognition of vehicle make and model," *Sensors*, vol. 16, no. 2, p. 226, 2016.

[13] S. Yu, Y. Wu, W. Li et al., "A Model for fine-grained vehicle classification based on deep learning," *Neurocomputing*, vol. 257, pp. 97–103, 2017.

[14] B. Hu, J.-H. Lai, and C.-C. Guo, "Location-aware fine-grained vehicle type recognition using multi-task deep networks," *Neurocomputing*, vol. 243, pp. 60–68, 2017.

[15] M. Lin, Q. Chen, and S. Yan, "Network in network," *Computer Science*, arXiv: 1312.4400, 2013.

[16] Y. Hou, L. Zhou, S. Jia, and X. Lun, "A novel approach of decoding eeg four-class motor imagery tasks via scout esi and cnn," *Journal of Neural Engineering*, vol. 17, no. 1, pp. 1–15, 2020.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1097–1105, 2012.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556, 2014.

[19] X. Dai, H. Yin, and N. K. Jha, "Nest: a neural network synthesis tool based on a grow-and-prune paradigm," *IEEE Transactions on Computers*, vol. 68, no. 10, pp. 1487–1497, 2019.

[20] F. Zhang, Y. Liu, Y. Zhou, Q. Yin, and H.-C. Li, "A lossless lightweight CNN design for sar target recognition," *Remote Sensing Letters*, vol. 11, no. 5, pp. 485–494, 2020.

[21] B. Ibrokhimov, C. Hur, and S. Kang, "Effective node selection technique towards sparse learning," *Applied Intelligence*, vol. 50, no. 10, pp. 3239–3251, 2020.

[22] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[23] X. Jin, L. Wu, X. Li et al., "Ilgnet: inception modules with connected local and global features for efficient image aesthetic quality classification using domain adaptation," *IET Computer Vision*, vol. 13, no. 2, pp. 206–212, 2019.

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

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] B. Liu, Q. Liu, Z. Zhu, T. Zhang, and Y. Yang, "Msst-resnet: deep multi-scale spatiotemporal features for robust visual object tracking," *Knowledge-Based Systems*, vol. 164, no. 15, pp. 235–252, 2019.

[27] Y. Jiang, Y. Li, and H. Zhang, "Hyperspectral image classification based on 3-D separable resnet and transfer learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 12, pp. 1949–1953, 2019.

[28] W. Shao, D. Pi, and Z. Shao, "A Pareto-based estimation of distribution algorithm for solving multiobjective distributed no-wait flow-shop scheduling problem with sequence-dependent setup time," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 3, pp. 1344–1360, 2019.

[29] H. Alaeddine and M. Jihene, "Deep network in network," *Neural Computing and Applications*, vol. 2020, Article ID 05008-0, 13 pages, 2020.

[30] X. Zhang and X. Zhang, "Global learnable pooling with enhancing distinctive feature for image classification," *IEEE Access*, vol. 8, pp. 98539–98547, 2020.

[31] W. Gong, H. Chen, Z. Zhang, M. Zhang, and H. Gao, "A data-driven-based fault diagnosis approach for electrical power dc-dc inverter by using modified convolutional neural network with global average pooling and 2-D feature image," *IEEE Access*, vol. 8, pp. 73677–73697, 2020.

[32] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[33] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, Y. Bengio et al., "Maxout networks," arXiv: 1302.4389, 2013.

[34] M. D. Zeiler and F. Rob, "Stochastic pooling for regularization of deep convolutional neural networks," arXiv:1301.3557, 2013.

[35] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," *Computer Science*, arXiv:1507.06228, 2015.

[36] E. Zhu and D. Pi, "Photovoltaic generation prediction of CCIPCA combined with LSTM," *Complexity*, vol. 2020, Article ID 1929372, 11 pages, 2020.

[37] W. U. Min, "Application of fuzzy neural network in network fault diagnosis," *Computer Knowledge and Technology*, vol. 14, 2019.

[38] P. Shamsolmoali, M. Zareapoor, and J. Yang, "Convolutional neural network in network (cnnin): hyperspectral image classification and dimensionality reduction," *IET Image Processing*, vol. 13, no. 2, pp. 246–253, 2019.

[39] E. Zhu, M. Xu, and D. Pi, "A novel robust principal component analysis algorithm of nonconvex rank approximation," *Mathematical Problems in Engineering*, vol. 2020, Article ID 9356935, 17 pages, 2020.

[40] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv:1207.0580, 2012.

[41] J. Wu, X. Zhu, C. Zhang, and P. S. Yu, "Bag constrained structure pattern mining for multi-graph classification," *IEEE Transactions on Knowledge and data engineering*, vol. 26, no. 10, pp. 2382–2396, 2014.

[42] J. Wu, S. Pan, X. Zhu, and Z Cai, "Boosting for multi-graph classification," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 416–429, 2015.

[43] C. Dai, D. Pi, J. Wu, L. Cui, and B. Johnson, "CenEEGs," *ACM Transactions on Knowledge Discovery from Data*, vol. 14, no. 2, pp. 1–25, 2020.

[44] C. Dai, J. Wu, D. Pi et al., "Brain EEG time-series clustering using maximum-weight clique," *IEEE Transactions on Cybernetics*, vol. 2020, pp. 1–15, 2020.

[45] J. Zhu, H. Shi, B. Song, S. Tan, and Y. Tao, "Deep neural network based recursive feature learning for nonlinear dynamic process monitoring," *The Canadian Journal of Chemical Engineering*, vol. 98, no. 4, pp. 919–933, 2020.