IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 24, 2018, accepted October 8, 2018, date of publication October 12, 2018, date of current version November 8, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2875525

A Novel Model Based on AdaBoost and Deep CNN for Vehicle Classification

WEI CHEN[1,2], QIANG SUN[1], JUE WANG[1], (Member, IEEE), JING-JING DONG[1], AND CHEN XU[1]

[1]School of Electronics and Information, Nantong University, Nantong 226019, China
[2]Medical School of Nantong University, Nantong 226001, China

Corresponding author: Chen Xu (xuchen@ntu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 6150164 and Grant 61771264, and in part by the Nantong University-Nantong Joint Research Center for Intelligent Information Technology under Grant KFKT2016B01 and Grant KFKT2017B04.

ABSTRACT Real-time vehicle classification is an important issue in intelligent transport systems. In this paper, we propose a novel model to classify five distinct groups of vehicle images from actual life based on AdaBoost algorithm and deep convolutional neural networks (CNNs). The experimental results demonstrate that the proposed model attains the highest classification accuracy of 99.50% on the test data set, while it takes only 28 ms to identify a vehicle image. This performance significantly outperforms the traditional algorithms, such as SIFT-SVM, HOG-SVM, and SURF-SVM. Moreover, the proposed deep CNN-based feature extractor has less parameters, thereby occupies much smaller storage resources as compared with the state-of-the-art CNN models. The high prediction accuracy and low storage cost confirm the effectiveness of our proposed model for vehicle classification in real time.

INDEX TERMS Real time, vehicle classification, CNN, AdaBoost, SVM.

I. INTRODUCTION

Vehicle classification in real time is extremely vital for many applications, especially in Intelligent Transportation Systems, ITS [1]. Accurate vehicle classification can provide accurate road information in time, and help monitoring road congestion, handling illegal conditions without delay, and reminding the driver of safety. It is undeniable that vehicle classification and recognition are the foundation and indispensible parts of ITS.

During the past decades, the technique of vehicle classification has been developed in a variety of research directions. For example, the technique has been investigated with the aid of magnetic induction coil, sensors, infrared ray, ultrasonic wave and radar [2]–[4]. Although a lot of efforts are being spent in improving these techniques, some shortcomings still exist. Firstly, the magnetic induction coil and the sensor are vulnerable and consumable. Secondly, radar, ultrasonic wave devices usually induce high implementation cost. With the development of digital video surveillance devices, visual image processing has become the major direction for study vehicle classification and recognition [5]–[7]. Compared with other technologies, visual image processing has the following merits: 1) It is simple to install and maintain digital video surveillance devices and it costs much less; 2) real-time video images are conducive to supervision, and the stored video data can be used for the subsequent analysis and processing. However, the vehicle characteristics acquired by video images could be easily affected by weather, illumination, angles, backgrounds, noises and other practical factors. As a result, the acquired vehicle images usually have low quality, as shown in Figure 1. How to perform accurate vehicle classification with low-quality images is a challenging problem, and this issue has captured considerable research attentions recently.

Based on images over the years, a host of researchers have been carried out on vehicle classification. In the next, we will briefly discuss the latest progress of these methods.

In general, vehicle classification based on images can be divided into two steps: extracting features and designing a classifier. Extracting features is the key to vehicle classification and recognition. There have been many conventional approaches to extract vehicle image features.

In the conventional approaches, the extracted features include contour texture features such as the vehicle length, width, ratio, and wheelbase [8] or moment features e.g. invariant moment [9], corner features [10], transformation domain characteristics [11], and the contourlet transform [12]. These extracted features are then manipulated with subsequent

FIGURE 1. Vehicle images from traffic road with low quality.

classifiers such as the SVM classifier and softmax classifier for vehicular classification. Although the accuracy of classification can be improved with more sophisticated features, it needs more time to classify the vehicle with these algorithms. Thus it is difficult to ensure real-time processing requirements.

In order to reduce the processing time, efficient feature selection and extraction have long been the focus of pattern recognition researchers. As an example, Cao *et al.* [14] proposed a method that combines boosting algorithm with linear support vector machine (SVM) to deal with low-altitude vehicle images, where the histogram of oriented gradient (HOG) features are extracted as feature descriptor of vehicle detection. This technique can reduce the influence of light to some extent, and it is also effective in describing the shape and appearance of the vehicles. Although a lot of efforts are being spent on improving the classification accuracy, the efficient and effective HOG descriptor has yet to be developed. On the one hand, the process of HOG descriptor still needs a large amount of time. On the other hand, it is sensitive to noise. Due to these limitations, the original method is not applicable to the actual complex road conditions and is unable to meet the requests for real time. Recently, Gabor filters have also been reported to achieve good performance in vehicle image classification. Arróspide and Salgado [13] improved the methods of vehicle recognition based on Gabor transform. In [13], combined with local binary mode method, log-Gabor filter and Gabor filter are proposed respectively. These two filters can be used to extract multi-scale features and directional characteristics of vehicles. Since the Gabor filter mainly extracts the texture features of vehicles, the classification result highly depends on the shooting angles. As a result, this method is sensitive to the edge of vehicle images and cannot adapt to the light intensity. This method is further improved in [15], where the scale-invariant feature transform (SIFT) algorithm is adopted to extract the key points of vehicles from multiple viewpoints, and best-bin first (BBF) method is used to find the matching feature points of the neighborhood to set corresponding relationship. SIFT not only has the invariance property in rotation, scale and brightness, but also maintains a certain stability for angle conversion and noise. What's more, the features can be quickly matched, so it can effectively improve the classification efficiency. The SIFT algorithm, however, still suffers from poor real-time performance. Yu *et al.* [16] adopts the method of graph theory to process image threshold, then inputs the gray values after secondary filtration into the back propagation (BP) neural network to update learning and identify vehicles. By comparison with the above methods, this approach has its advantage in extracting features adaptively. Nevertheless, it still has the problems of local minimum and large training parameters.

In recent years, deep learning technology has developed rapidly. Deep CNN has been applied in the area of image classification, e.g. the ImageNet LSVRC [17]–[19]. Deng and Wang [20] proposed a vehicle recognition system by adopting a deep CNN-based feature extraction method and SVM classifier. In this paper, three major types of vehicles on the highway can be effectively recognized, showing significant improvements in both accuracy and speed by using the new technique. Inspired by these latest approaches, the aim of this paper is to provide methods to further improve the accuracy of real time vehicle classification.

The main contributions of this research are summarized as follows:

1. In comparison with SIFT-SVM, HOG-SVM and SURF-SVM methods, the proposed model achieves the highest accuracy about 99.50%, even under adverse circumstances where low quality data are acquired in the night, rain, and haze conditions.
2. We propose a high-efficiency deep CNN model to extract the vehicle image features. The proposed feature extractor can dramatically save the cost of storage resources and training time.
3. The proposed method reduces the computation cost while ensuring high accuracy. After fine training,

the operating time of each picture is only about 28 ms, which adequately meets the needs of real-time processing.

The rest of the paper is organized as follows: Section II describes the proposed model for vehicle classification in detail, including the architecture of the frame work and the feature extractor. Then the detailed principle about how it is suitable for vehicle classification is introduced in this section. Section III presents the experiment results and analysis, as well as comparison with other algorithms. Lastly, conclusions are concluded in Section IV.

II. THE PROPOSED MODEL

In this section, we describe the proposed model in detail. First of all, we provide an overview of the architecture and introduce how it is suitable for vehicle classification. Then, the CNN-based feature extractor is described. Each layer of CNN is elaborated in this section. After that, we present the algorithms of AdaBoost and SVM especially the one versus one method. At last, the network training schedule is given in detail.

A. FRAMEWORK OVERVIEW

Figure 2 depicts the proposed model architecture. It begins with an input dataset, which accepts the size of $224 \times 224 \times 3$ pixels-sized images (224 is the pixel of height and weight of vehicle images, and 3 is the number of image channels). In order to reduce the cost of storage and training time, we designed a CNN model with 12 convolutional neural layers and 2 fully connected layers (FC) as a feature extractor. This CNN model has less parameters and more deep convolutional layers. As shown in Figure 2, the blue module accepts the dataset to train the CNN model, which is described in the next section. After training, the CNN model can be used to extract useful and highly representative features of vehicle images. In this novel model, the CNN is used as a feature extractor, and SVM is used as the weak classifier of the AdaBoost. The proposed model training procedure is described in Figure 3.

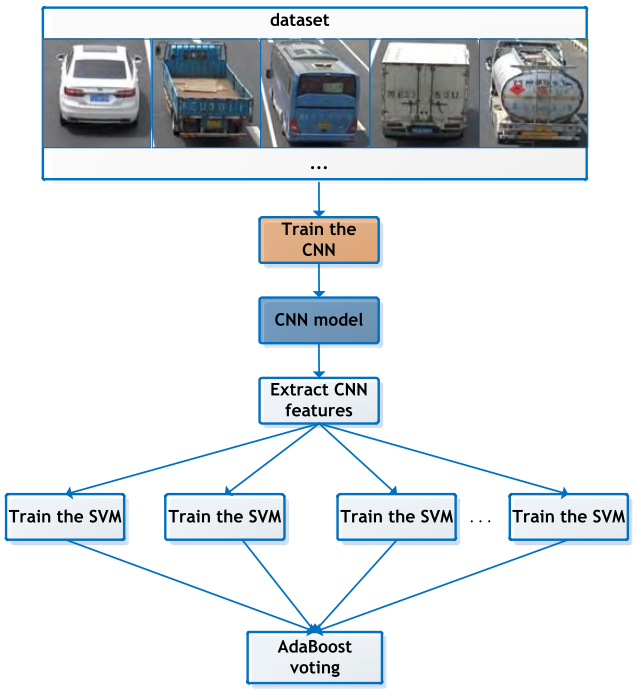


FIGURE 2. The proposed model for vehicle classification.

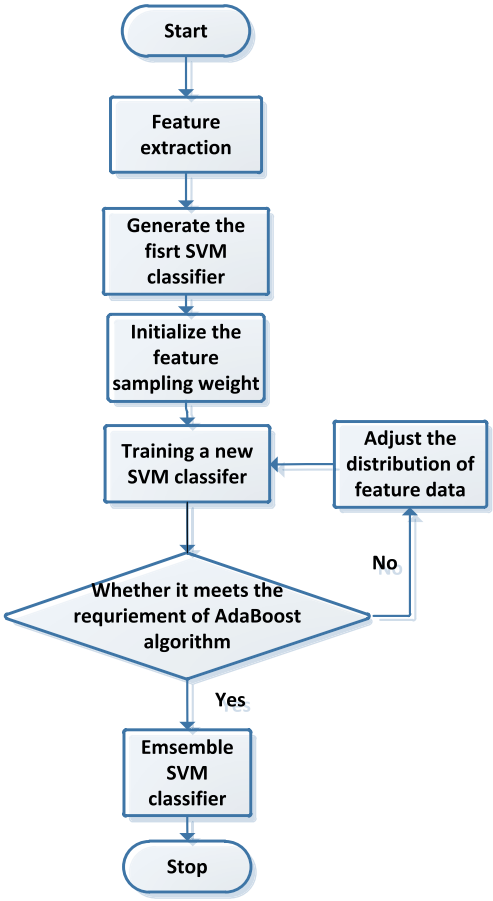


FIGURE 3. The model training procedure.

B. THE CNN-BASED FEATURE EXTRACTOR

Being a multi-layer neural network, CNN can automatically obtain feature representation from input data. Figure 4 describes the architecture of the proposed CNN-based feature extractor. This structure is based on the design principle of VGGNet [21] and AlexNet [18]. The entire architecture includes 6 convolution blocks and 2 fully connected (FC) layers, and each convolutional block is followed by a max-pooling layer which leads to the output resolution reduced by a half. Each convolutional block is comprised of two convolutional layers followed by a Batch Normalization (BN) layer and a Rectification layer (ReLU) activation function. We will describe the detailed introduction about linear convolutional layer and nonlinearity ReLU in the following section.

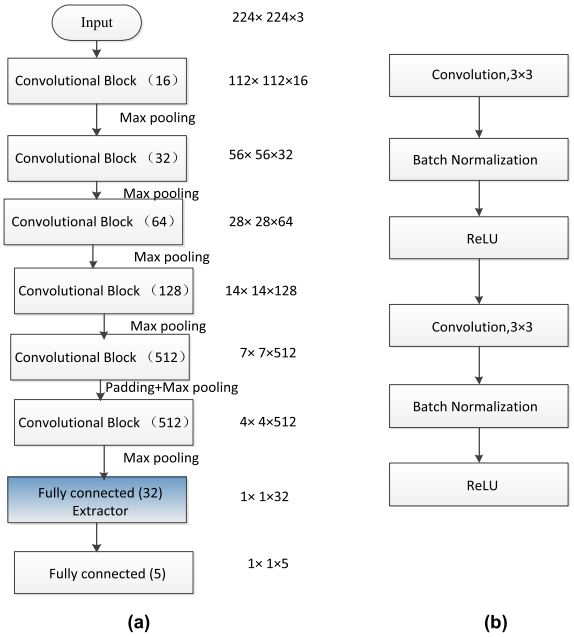


FIGURE 4. The proposed CNN-based feature extractor. (a) CNN architecture. (b) Convolutional block.

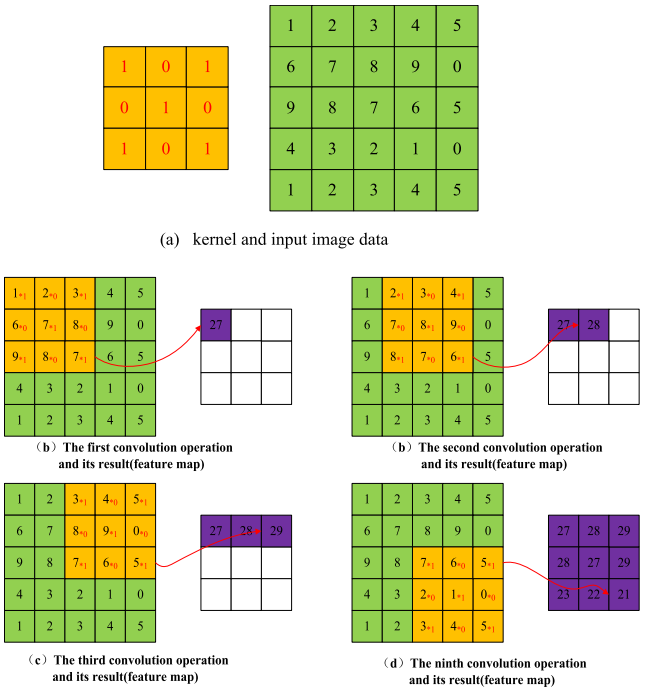


FIGURE 5. An example of image convolution. (a) kernel and input image data. (b) The second convolution operation and its result(feature map). (c) The third convolution operation and its result(feature map). (d) The ninth convolution operation and its result(feature map).

In the feature extractor, we use 6 stacking convolutional blocks with small kernel size (3×3) to increase the depth of the network. In the last but one layer we use 32 dimensional fully connected neurons to extract the features of the vehicle images. The final FC layer is connected to the number of classes in the dataset. In this case, the number of classes is 5. Therefore, it has 5 neurons of fully connected layer. As shown in figure 4 (a), in the right side, we give the output size of each module. The aim of the proposed CNN model is to enhance the effectiveness of the neural network to a certain extent, and reduce the required parameters in the structure.

In summary, the intended CNN model consists of 20 layers with 12 convolution layers, 6 max pooling layers and 2 fully connected layers. In order to accelerate the training of the deep learning model, Batch Normalization is adopted in the convolutional block. After the features being extracted by the CNN model, we further adopt a trained SVM classifier for vehicle classification.

Having introduced the structure of the feature extractor, complete details of some key modules in our networks will be provided at a later time.

1) CONVOLUTION LAYER

Convolution layer is the basic component in convolution neural networks. It is composed of multiple feature surfaces (feature maps), each of which is composed of many neurons. The neurons are connected by the convolution kernel to the local region of the upper feature surfaces. The convolution layer of CNN can extract different features of the input by the convolution operation. By increasing the depth of the convolution layers, more advanced features can be extracted.

Representing the input image by "I", and the two-dimensional convolution kernel by "K"; the convolution of the input image is:

$$C(i,j) = (I * k)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \qquad (1)$$

In (1), convolution must reverse the convolution kernel and then aggregate the weights. Once the convolution is done, the convolution kernel moves a stride. In our research, 3×3 convolution kernel is used in the CNN model. Figure 5 explains an example of a convolution operation with 3×3 kernel size. In this instance, the input is an image data presented by a 5×5 matrix and the stride is 1. After the ninth convolution operation, we can obtain the 3×3 feature maps. It can reduce the size of input images of each layer, and extract some useful global and local features.

2) MAX-POOLING LAYER

Max-pooling has been used for the operation of feature mapping units of hidden neurons after convolution layers, owing to its suitability and better effects than sigmoid function. On account of better ability of texture feature extraction, it has been applied to the presented structure to replace mean-pooling. It is closer to biological characteristics, hence it can well relieve the problems of over-fitting and vanishing gradient to some extent. The number of parameters which needs to be learned is dramatically reduced by pooling operations while the key features of the vehicle with invariance under translation in space are preserved. The relevant formula can

be depicted in (2) and (3).

$$y_i = max_{R\times R}\left\{y_i^{r\times r}\right\} f\,(r, r) \tag{2}$$

$$\mathrm{f\,(r,\,r)} = \varepsilon \cdot y_i^{k-1} \times \omega_{i,j}^{k} + e_j^{k} \tag{3}$$

In the above formulas, $\mathrm{max_{R\times R}}$ denotes max-pooling operate in a R × R region, $\mathrm{y_i^{r\times r}}$ is the i-th output maps of a r × r window, and f (r, r) represents the window function of setting blocks where ε is a trainable variant.

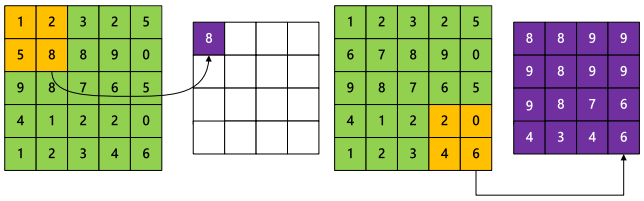


FIGURE 6. Illustration of the max-pooling operation.

Figure 6 shows an example of the max-pooling operation. The process of the picture from left to right in turn is: the first max-pooling, the feature map after first max-pooling, the sixteen max-pooling and the feature map after the sixteen max-pooling. It is clear seen that the result of the confluence operation is smaller than its input image. In fact, the confluence operation is actually a "drop sampling" operation. The pool layer has three functions: feature invariance, feature dimensionality reduction, and prevention of over-fitting.

3) DROPOUT LAYER

Over-fitting is such a considerable issue in deep learning, that various methods have been designed by researchers. In order to handle this issue, two approaches are adopted in our CNN-based feature extractor. For one thing, data augmentation was adopted in the training process such as image horizontally flipping and random crops. For another thing, the dropout layer proposed by Hinton *et al.* [22] is adopted in the feature extractor. The dropout layer can make the network structure cleaner and more regularized. The correlation between neurons decreases after convolution operation and the network can get better parameters in the process of updating weights. The specific working process is shown in Figure 7.

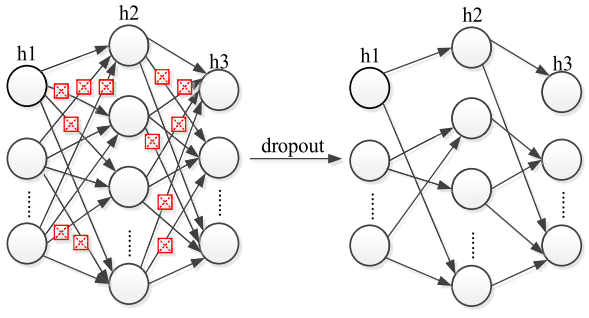


FIGURE 7. The diagram of the dropout layer.

In the deployment of CNN, we add the dropout layers into the two FC layers. The rates of all dropout layers are set to be 0.3.

4) ACTIVATION FUNCTION

The activation function layer is also called the nonlinearity mapping layer. It is used to increase the nonlinear expressive power of the whole network. The stacking of some linear operation layers can only play the role of linear mapping, but cannot form complex functions. There are several available activation functions to be chosen. In the following, we will introduce the rectified linear unit (ReLU) that we used in the proposed network.

ReLU is one of the most commonly used activation functions in deep convolution neural networks [23]. In the process of error back propagation, it's difficult or even impossible to transmit the error of the derivative in the region to the front layer, which leads to training failures. The ReLU function is a piecewise function, defined as:

$$Rectifier\,(x) = \{max(0, x)\} = \begin{cases} x & if\ x > 0 \\ 0 & if\ x < 0 \end{cases} \tag{4}$$

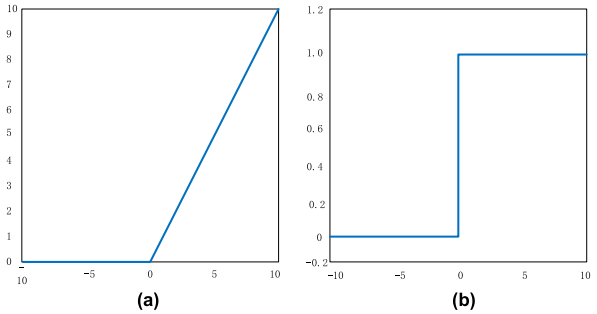


FIGURE 8. ReLU function and its function gradient. (a) ReLU function. (b) ReLU gradient function.

As shown in Figure 8, when x is greater than 0, the function gradient is 1 or 0. It is also found that the function can help the stochastic gradient descent (SGD) method converge, and the convergence rate is about 6 times than Sigmoid function [23].

C. TRAINING PROCESS OF CNN

The training stage of our CNN is shown in Table 1. We train the proposed model using standard hinge loss function instead of softmax cross entropy loss function for optimization. Adaptive Moment Estimation (Adam) is adopted for learning the gradient reduction, which is the most popular optimization approach for neural network training. In the proposed model, L2 regularization was computed for each of the fully connected layers' weights and bias matrices was applied to the eventual loss function. In order to prevent over-fitting, we applied dropout to fully connected layers. Table 2 shows the values of initial parameters of Adam algorithm. After the CNN model being trained, the output layer of the CNN is then removed. Then we use the trained CNN model to extract features from the raw vehicle images.

TABLE 1. Training stage of the CNN.

Algorithm 1 Train CNN with the learning algorithm of Adam

Initialization:

$Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(n)}\}$ is marked as dataset; v is the initial velocity; m is the size of random sampling; v and r is the first moment vector and second moment vector; $f(x; w)$ is the convolution network; α is the initial learning efficiency; β_1 is the parameter of momentum decay; β_2 is the parameter of learning efficiency decay; w is the parameter of initial weight.

Training:

For t<k

1. "m" data are randomly selected as samples

2.Calculate the gradient of the current sample data

$$g = \frac{1}{m}\sum_{j=1}^{m}\frac{\partial L(y^{(j)}, f(x^{(j)}; w))}{\partial_w}$$

3. Update the current velocity: $v = \beta_1 \cdot v + (1-\beta_1)g$

4.Update the current learning efficiency: $r = \beta_2 \cdot r + (1-\beta_2)g^2$

5. Update the frequency of training: $t = t + 1$

$$vb = \frac{v}{1-\beta_1^t}, \quad rb = \frac{r}{1-\beta_2^t}$$

6.Update parameter: $w = w - \frac{\alpha}{\sqrt{rb}+\delta}vb$

TABLE 2. Initial parameters of Adam.

Initial parameters	Value
v	0
r	0
α	0.001
δ	10^{-8}
β_1	0.9
β_2	0.999

D. BASE CLASSIFIER

In our model, we use SVM as the base classifier due to its better classification effect. SVM is a machine learning model based on statistical learning theory and structural risk minimization principle. Figure 9 illustrates the SVM for binary classification. Two classes are represented respectively by blue circles and red squares. Many linear classifiers can separate these two classes, among which the maximum separation is called the optimal separation hyperplane [24]. The purpose of SVM is to use available training samples to establish the optimal hyperplane in space. The principle of SVM is also described in Figure 9.

As for a binary classification problem that the data $x_i, i = 1, 2, \ldots, n.$, belongs to either class I or class II, both of which are labeled as $y_i = \pm 1$, the decision function is given in (5).

$$f(x) = sign(\omega \cdot x + b) \tag{5}$$

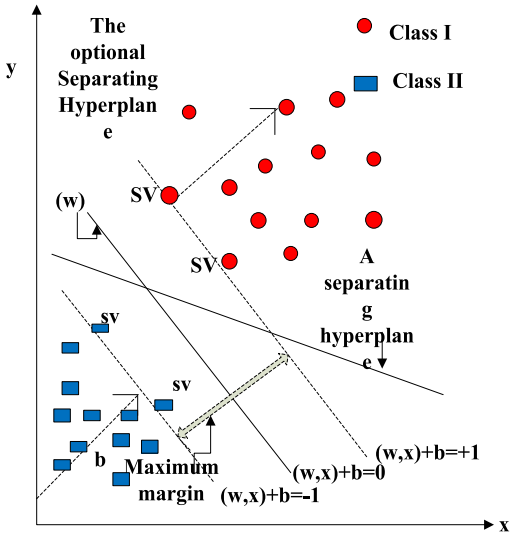


FIGURE 9. Illustration of binary SVM.

For the linearly separable case, the following condition should be satisfied:

$$y_i(\omega \cdot x + b) > 0 \forall_i \tag{6}$$

In (6), ω defines a normal vector, the hyperplane denotes the boundary, and b (bias) represents the hyperplane's distance from the origin data. The optimal hyperplane aims to maximize the margin, which refers to the distance between the hyperplane and the nearest data point of each class. These points are called support vectors that lie on the margin or within the margin. Consequently, the problem of constructing optimal hyperplane is transformed into the quadratic optimization problem in (7).

$$min,\ \Phi(w) = \frac{1}{2}\|w\|^2 \tag{7}$$

$$\text{Subject to}: y_i(w \cdot x_i + b) \geq 1, \quad i = 1, 2, \ldots, n \tag{8}$$

Because some available data can be non-linear and non-separable, training data do not always satisfy the constraint conditions. Therefore, in order to find the optimal hyperplane, the constraint optimization problems can be solved in (9).

$$min,\ \phi(w) = \frac{1}{2}\| w \|^2 + \sum_{i=1}^{n}\varepsilon_i \tag{9}$$

$$\text{Subject to: } y_i(w \cdot x_i + b) + \varepsilon_i \geq 1\varepsilon_i \geq 0, \quad i = 1, 2, \ldots n \tag{10}$$

In (10), the variable ε_i defines the positive slack variables, which allows misclassification for some data points when necessary. The variable C defines the generalization parameter or soft margin classifier, which is a trade-off between the misclassification and boundary complexity. In order to deal with the high number of attributes of data examples, the problem with complex constraint can be easily converted

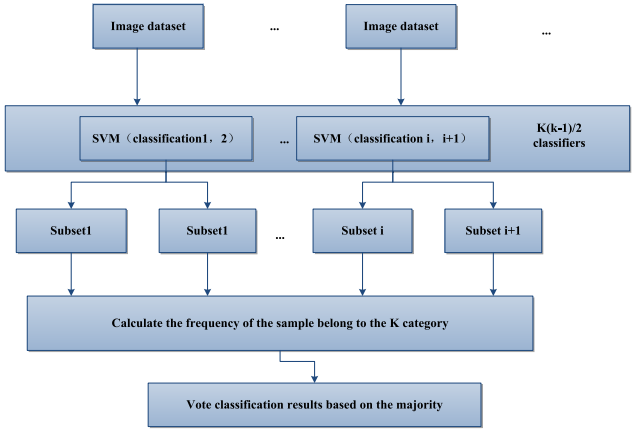


FIGURE 10. The base classifier based on SVM.

into the equivalent Langrange dual problem. The problem becomes:

$$\text{Max}\{D(\alpha)\} = \sum_{k=1}^{n} \alpha_k - \frac{1}{2} \sum_{i,j=1}^{n} \left\{ y_i \alpha_i y_j \alpha_j \left(x_i, x_j \right) \right\} \tag{11}$$

Constraint:

$$\sum_{j=1}^{n} y_j \alpha_j = 0 \quad \text{and } \alpha_i \geq 0, \ i = 1, 2, \ldots n \tag{12}$$

Kernel tricks are able to address non-linear case. $\text{k}\left(\text{x},\text{x}'\right)$, kernel function, is used to map the data from an input space to a higher dimension space by using a non-linear mapping function, $\emptyset(x)$. Then, within the higher dimension space, a linear optimal separating hyperplane can be constructed for separating two classes [24], [25]. The decision function will be:

$$\text{f}(\text{x}) = \sum_{i=1}^{m} \{ y_i \alpha_i K(x, x_i) \} + b \tag{13}$$

Our problem is a multiple classification problem. Nevertheless, it can be solved by decomposing SVM into several binary problems. For this issue, the three prime approaches can be summarized: one-versus-all, one-versus-one and direct acyclic graph. Among these methods, one-versus-one method is most effective for the practical situations due to its small training period and good generalization ability [26]. For this reason, one-versus-one method is adopted in our study. For k-event, we can use one-versus-one method to construct N binary SVMs.

$$N = \frac{k(k-1)}{2} \tag{14}$$

In this paper k is set to be 5, then N equals 10. Each SVM is trained on the basis of the dataset of two classes. The implementation process of our base classifier is shown in Figure 10.

E. ENSEMBLE SCHEME

In the proposed model, we use AdaBoost algorithm to assemble SVM methods. AdaBoost, proposed by Freund and Schapire in 1995, combines a large set of weak classification function. It uses a weighted majority vote, and associates a larger weight with good classification functions and a smaller weight with poor classification functions. In this paper, SVM was assembled by AdaBoost for ensemble learning. Figure11 shows the structure of ensemble scheme.

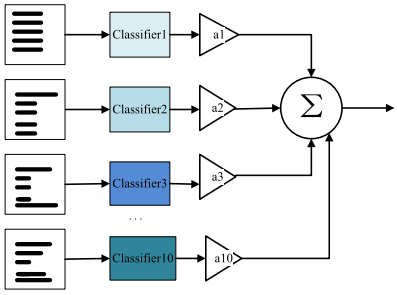


FIGURE 11. The structure of ensemble scheme.

In Figure 11, the left-most part of the graph represents the features extracted from CNN, where the different widths of the histogram indicate the different weights of each sample. Based on the previous analysis, we utilize 10 SVM classifiers to assemble the ensemble scheme. Once the first classifier is processed, the prediction result is weighted by a value in the triangle. The weighted summation of the output of each triangle yields the output result finally. Table 3 gives the process of the ensemble learning.

F. SUMMARY

In our proposed architecture, we first use a high-efficiency CNN model to extract vehicle image features then, its output layer SVM is assemble by AdaBoost for ensemble learning.

III. EXPERIMENT AND ANALYSIS

To analyze the performance of our proposed model, we designed experiment and compare the proposed model with some state-of-the-art methods on the same dataset. The dataset used in our experiment is introduced firstly. Then, the hardware and software of the experiment are described in detail. Lastly, we elaborate the results of the proposed method and compare it with the state-of-the-art methods.

A. DATASET

Taking the traffic safety into account, we captured the images of vehicles' rears in actual researches. Therefore, the vehicle rear images were used as dataset in the experiments. Altogether, there are five different kinds of vehicles, including cars, trucks, vans, buses and tractors. These vehicle images are from two different ways. Both of the two are introduced in detail as follows:

1) 23510 vehicle images are downloaded from CompCars database [25], which can be utilized for a multitude

TABLE 3. The process of the ensemble learning.

Step 1 Set the data and parameter:
$T = \{(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)\}$, is marked as input dataset, where$x_i \in X \subseteq R^n, y_i \in Y = 0,1,2 \ldots C$ (C is the number of our classify).
M is the max number of iteration. T is the number of the weak classifier.

Step 2 Initialize the weight distribution of the training sample:

$$D_1 = (w_{1,1}, w_{1,2}, \ldots, w_{1,i}), w_{1,i} = \frac{1}{N}, i = 1,2, \ldots, N$$

Step 2 Update and learn the weight:
For m=1,2,3…,M
1)Use the training dataset with the weight distribution of $D_m(x)$

2）Calculate the error rate of classifier of $G_m(x)$on training dataset:

$$e_m = \sum_{i=1}^{N} w_{m,i} I(G_m(x_i) \neq y_i)$$

3）Calculate the weight of $G_m(x)$ in a strong classifier:

$$a_m = \frac{1}{2} log \frac{1 - e_m}{e_m}$$

4）Update the weight distribution of training data, where z_m is normalization facto):

$$w_{m+1}, i = \frac{w_{m,i}}{z_m} \exp(-a_m y_i G_m(x_i)), i = 1,2, \ldots, T$$

$$z_m = \sum_{i=1}^{N} w_m, \exp(-a_m y_i G_m(x_i))$$

Step 3 Combine the final strong classifier

$$F(x) = sign(\sum_{i=1}^{N} a_m G_m (x))$$

of computer vision algorithms. This dataset contains diverse car images distributed in all angles, including front, rear, side, front-side and rear-side. We only used the rear images in our experiment.

2) In order to increase the size of the dataset, we took pictures of vehicles on an overpass bridge near Sutong Bridge. As shown in the Figure 12, we change the size of camera angles β to obtain images from different angles. The number of images from the real-word is 21720.

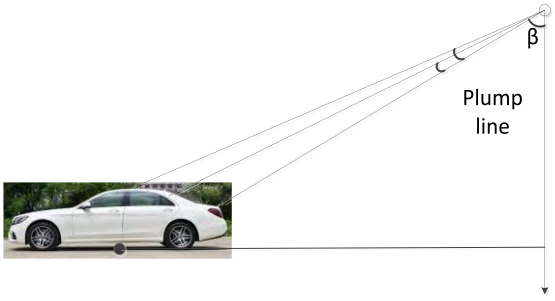


FIGURE 12. Photographing vehicles from different angles.

Figure 1 shows some vehicle images from traffic road. Some of them are captured from bad weather such as night, rain, and haze, which make the vehicle classifying task still a big issue. In Figure 1, vehicles from the first row to the last in turn are cars, trucks, vans, buses and tractors. Detail information of training and testing images of the five types

TABLE 4. Numbers of training and testing images for the five types of vehicles.

Type	Number of Training Images	Number of Test Images	Total
Car	8000	2000	10000
Truck	7680	1920	9600
Van	7440	1860	9300
Bus	7920	1980	9900
Tractor	5144	1286	6430
All	36184	9046	45230

of vehicles are demonstrated in Table 4. As shown in Table 4, 36184 images are set for training dataset, and 9046 images are set for test dataset. The proportion of training, validation dataset and testing dataset in total samples is 80%, 20%, 20%, respectively. It should be noted that a part of the dataset are acquired under bad weather conditions.

B. HARDWARE AND SOFTWARE

The experiments and the comparison experiments are performed in the same environment. The hardware and software used in the experiment are listed in Table 5. We use the Caffe platform which is an efficient framework for deep learning to train and optimize the model [26]. It only took about 8 minutes to train the CNN model on 36184 training images with 10000 epochs. The training work was implemented offline (before employing the model for vehicle classification).

TABLE 5. Hardware and software.

Item	Content
Processor	Intel Core i7-7700k(CPU)with 4.20GHz,
GPU	GEFORCE GTX 1080 Ti
Memory	64.00GB
Operating system	Ubuntu 16.04
Python	Python 2.7
Cuda	CUDA 9.0
Cudnn	CUDNN 7.5

C. EXPERIMENTAL RESULTS

1) PERFORMANCE OF THE CNN EXTRACTOR

Learning curve is one of the most crucial evaluation measures for the deep learning algorithm. As can be seen in figure 13, with the proposed model, both the training loss and validation loss continue to decrease as the number of epochs goes up. From this curves, we can draw the conclusion that our model didn't incur an "over-fitting" or "under-fitting" phenomenon.

Besides, classification accuracy is also an imperative aspect for evaluating the classifier in machine learning. It is defined as the number of correctly classified vehicles divided by the total number of testing dataset. We use the accuracy of confusion matrix diagram to analyze the proposed method. As shown in figure 14, the actual prediction accuracy of these five vehicles is 99.80%, 99.74%, 99.62%, 98.85% and

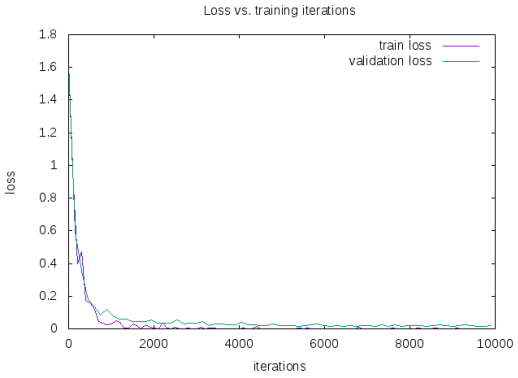


FIGURE 13. Training loss and validation loss curve.

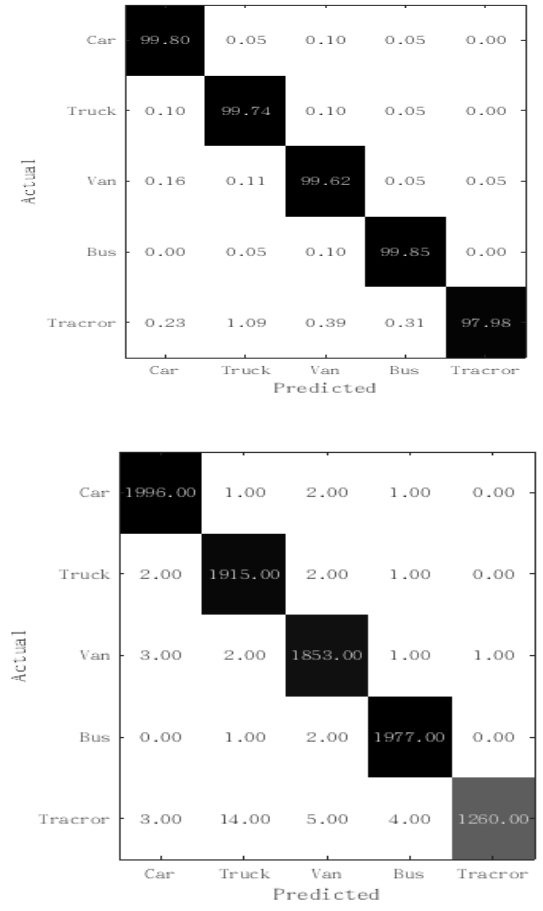


FIGURE 14. The accuracy of confusion matrix diagram by our method.

97.98%, respectively. The total classification accuracy on testing dataset is 99.50%. Since the shape of the car is obviously different from the other four types, prediction accuracy of the car is higher than other vehicles. While the similarity between the tractor and the truck is higher, there is a 1.09% chance of a tractor being misjudged as a truck. As a result, 14 tractors are misjudged as trucks. The misjudged rate of van (from left to right) is 0.16%, 0.11%, 0.05% and 0.05%, respectively. From the confusion matrix, it can be found that three vans are misjudged as cars, two vans are misjudged as trucks, one van is misjudged as a bus and one van is misjudged as a tractor. Other vehicles have the same analysis process.

TABLE 6. Performance comparison between the proposed method and other deep CNN model.

Model	Training Time(minute)	Volume (MB)	Predicting Time(ms)	Accuracy
AlexNet	34	240	20.6	97.06%
VGGNet-16	51	552	410.9	98.41%
GoogLeNet	38	51	44.4	99.03%
Our method	**8**	**25**	**28**	**99.50%**

2) COMPARISON OF DIFFERENT METHODS

In the experiment, some state-of-the-art deep convolutional neural networks including AlexNet [18], VGGNet-16 [21] and GoogLeNet [27] are compared with the our method. These networks are trained on the same training dataset with the same hyper-parameters. We made the comparisons from four aspects: the accuracy, the training time, the predicting time of testing a vehicle image and the volume of the model. Detailed comparison results for these methods are listed in Table 6. From Table 6, it can conclude that these state-of-the-art models can achieve more than 97% accuracy, but the proposed CNN model takes only 8 minutes to train. Furthermore, our method attains the highest accuracy, and occupies smallest storage resources (25MB). Although the time required to predict a vehicle image is not the least by the proposed method, it does not affect the real-time application requirements. In general, the proposed model achieves the highest classification accuracy and also reduces the required hardware devices.

Having compared with various models of deep convolutional neural networks, some traditional feature extraction methods, such as the HOG method [28], [29], SIFT method [15] and SURF method [29] that we also call hand-crafted feature methods are also compared in this paper. The comparison results are shown in Table 7. The proposed method has the highest accuracy than the traditional artificial feature extraction methods, and the accuracy rate reaches 99.50%, about 13% higher than the HOG+SVM method, 6.2% higher than the SIFT+SVM method and 6.3% higher than SURF+SVM method.

The proposed feature extractor based on CNN can automatically learn the features without preprocessing the image, and can directly handle the RGB original image. However, the hand-crafted feature methods need several of image preprocessing steps, such as image transformation, enhancement, gray scale, and so on. Based on the above analysis, the operating time by the proposed method is much less than others. To be exact, the average time spent in predicting a picture is about 28 ms, which can adequately meet the needs of real-time processing.

Besides, we compared the proposed method with above mentioned methods on the same testing dataset. It is clearly

TABLE 7. Results of testing dataset by the proposed method and other methods.

Method	Accuracy of Testing	Predicting Time (ms)
SIFT+ SVM	90.82%	6530
HOG + SVM	86.3%	10270
SURF+ SVM	93.05%	450
Our proposed method	**99.50%**	**28**

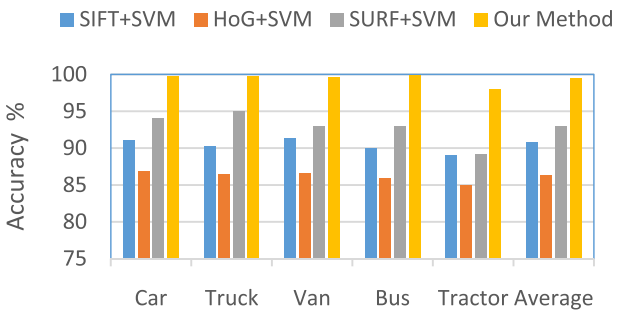


FIGURE 15. Performance of comparison with different methods in terms of accuracy for testing dataset.

seen that our method outperforms the comparative methods in 5 type vehicle images as shown in Figure 15.

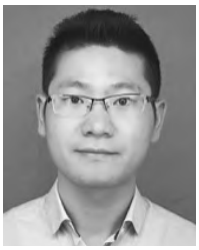
IV. CONCLUSIONS

A novel model based on AdaBoost algorithm and deep convolutional neural networks has been proposed for vehicle classification in this paper. Inspired by VGGNet and AlexNet a high-efficiency deep CNN model is designed to directly extract the features of vehicle images. The output layer of the CNN is taken as the base learner of the AdaBoost algorithm. Compared with the state-of-the-art methods, the proposed method has many advantages. First of all, the presented model is able to perform the highest accuracy even in low quality data. In addition, the proposed feature extractor can dramatically save the cost of storage resources. Last but not least, the proposed method reduces the computation cost while ensuring high accuracy. Our method can be used in many fields such as intelligent transport systems and traffic supervision in real time.

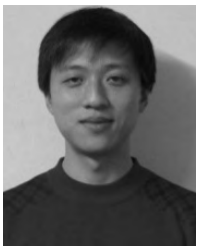
Future work is currently being carried out to further study in the following aspects. First, we will enrich the vehicle type and its images to augment the deep learning datasets. Second, we will also design a system to put this method into the practical application.

REFERENCES

[1] Y. Tang *et al.*, "Vehicle detection and recognition for intelligent traffic surveillance system," *Multimedia Tools Appl.*, vol. 76, no. 4, pp. 5817–5832, 2015.
[2] A. MocholíÂ-Salcedo, J. H. Arroyo-Núñez, V. M. Milián-Sánchez, G. J. Verdú-Martín, and A. Arroyo-Núñez, "Traffic control magnetic loops electric characteristics variation due to the passage of vehicles over them," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1540–1548, Jun. 2017.
[3] Y. Jo and I. Jung, "Analysis of vehicle detection with WSN-based ultrasonic sensors," *Sensors*, vol. 14, no. 8, pp. 14050–14069, 2014.
[4] K. Mu, F. Hui, and X. Zhao, "Multiple vehicle detection and tracking in highway traffic surveillance video based on sift feature matching," *J. Inf. Process Syst.*, vol. 12, no. 2, pp. 183–195, 2016.
[5] N. C. Acheruvu and V. Muthukumar, "Video based vehicle detection and its application in intelligent transportation systems," *J. Transp. Technol.*, vol. 2, no. 4, pp. 305–314, Oct. 2012.
[6] K. Robert, "Video-based traffic monitoring at day and night vehicle features detection tracking," in *Proc. ITSC*, St. Louis, MO, USA, Oct. 2009, pp. 1–6.
[7] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, "Vehicle detection and tracking in car video based on motion model," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 583–595, Jun. 2011.
[8] X. Yuan, Y.-J. Lu, and S. Sarraf, "Computer vision system for automatic vehicle classification," *J. Transp. Eng.*, vol. 120, no. 6, pp. 861–876, 1994.
[9] Q. Tian, T. Zhong, and H. Li, "A new method for vehicle detection using MexicanHat wavelet and moment invariants," in *Proc. SiPS*, Taipei City, Taiwan, Oct. 2013, pp. 289–294.
[10] M. Liu, C. Wu, and Y. Zhang, "Motion vehicle tracking based on multi-resolution optical flow and multi-scale Harris corner detection," in *Proc. IEEE Int. Conf. ROBIO*, Sanya, China, Dec. 2007, pp. 2032–2036.
[11] M. A. Naiel, M. O. Ahmad, and M. N. S. Swamy, "Vehicle detection using approximation of feature pyramids in the DFT domain," in *Proc. Int. Conf. Image Anal. Recognit.*, vol. 9164, Jul. 2015, pp. 429–436.
[12] S. Rahati, R. Moravejian, E. M. Kazemi, and F. M. Kazemi, "Vehicle recognition using contourlet transform and SVM," in *Proc. 5th Int. Conf. Inf. Technol.*, Las Vegas, NV, USA, Apr. 2008, pp. 894–898.
[13] J. Arróspide and L. Salgado, "Log-Gabor filters for image-based vehicle verification," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2286–2295, Jun. 2013.
[14] X. Cao, C. Wu, P. Yan, and X. Li, "Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos," in *Proc. 18th IEEE Int. Conf. Image Process.*, Brussels, Belgium, Sep. 2011, pp. 2421–2424.
[15] L. Hua, W. Xu, T. Wang, R. Ma, and B. Xu, "Vehicle recognition using improved SIFT and multi-view model," *J. Xi'an Jiaotong Univ.*, vol. 47, no. 4, pp. 92–99, 2013.
[16] W. Yu, L. Lei, and S. Feng, "A new vehicle recognition approach based on graph spectral theory and BP neural network," in *Proc. Int. Conf. Comput. Sci. Electron. Eng.*, Hangzhou, China, Mar. 2012, pp. 84–86.
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[19] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[20] L. Deng and Z. Wang, "Deep convolution neural networks for vehicle classification," *Appl. Res. Comput.*, vol. 33, no. 3, pp. 930–932, 2016.
[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition,"in *Proc. ICLR*, 2015, pp. 1–14.
[22] G. Hinton *et al.*, "Improving neural networks by preventing co-adaptation of feature detectors," *Comput. Sci.*, vol. 3, no. 4, pp. 212–223, Jul. 2012.
[23] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2012, pp. 315–323.
[24] P. Gangsar and R. Tiwari, "Comparative investigation of vibration and current monitoring for prediction of mechanical and electrical faults in induction motor based on multiclass-support vector machine algorithms," *Mech. Syst. Signal Process.*, vol. 94, pp. 464–481, Sep. 2017.
[25] L. Yang, P. Luo, C. C. Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Proc. IEEE Conf. CVPR*, Boston, MA, USA, Jun. 2015, pp. 3973–3981.
[26] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[27] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. CVPR*, Boston, MA, USA, Jun. 2015, pp. 1–9.
[28] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "A hybrid vehicle detection method based on viola-jones and HOG + SVM from UAV images," *Sensors*, vol. 16, no. 8, p. 1325, 2016.
[29] J.-W. Hsieh, L.-C. Chen, D.-Y. Chen, and S.-C. Cheng, "Vehicle make and model recognition using symmetrical SURF," in *Proc. 10th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Krakow, Poland, Aug. 2013, pp. 472–477.



WEI CHEN is currently pursuing the Ph.D. degree with the School of Electronics Information, Nantong University, China. His research areas mainly include computer vision and deep learning.



QIANG SUN was born in Nantong, China. He received the Ph.D. degree in communications and information systems from Southeast University, Nanjing, in 2014. He was a Visiting Scholar with the University of Delaware, USA, in 2016. He is currently an Professor with the School of Electronics and Information, Nantong University, Nantong. His research interests include deep learning and wireless communication.

JUE WANG (S'10–M'14) received the B.S. degree in communications engineering from Nanjing University, Nanjing, China, in 2006, and the M.S. and Ph.D. degrees from the National Communications Research Laboratory, Southeast University, Nanjing, in 2009 and 2014, respectively.

From 2014 to 2016, he was with the Singapore University of Technology and Design as a Post-Doctoral Research Fellow. He is currently a Lecturer with the School of Electronic and Information Engineering, Nantong University, Nantong, China.

Dr. Wang has served as a technical program committee member for a number of IEEE conferences, and a reviewer for several IEEE journals. He was as an Exemplary Reviewer of the IEEE TRANSACTIONS ON COMMUNICATIONS for 2014.

His research interests include MIMO wireless communications, multiuser transmission, MIMO channel modeling, massive MIMO systems, physical layer security, and deep learning.

JING-JING DONG received the B.S. degrees from Nantong University in 2016. He is currently pursuing the M.S. degree with the School of Electronics Information, Nantong University, China. His research areas mainly include image procession and convolution neural networks.

CHEN XU was born in Nantong, China. He was the Dean of the Key Laboratory of ASIC Design of Jiangsu Province, Jiangsu, China. He is currently a Professor with the School of Electronics and Information, Nantong University, Nantong. His research interests primarily comprise machine learning, image processing, intelligent transportation systems, medical information processing, and wireless communication.