Hindawi
Advances in Multimedia
Volume 2018, Article ID 3138278, 10 pages
https://doi.org/10.1155/2018/3138278

Research Article

Pretraining Convolutional Neural Networks for Image-Based Vehicle Classification

Yunfei Han,[1,2,3] **Tonghai Jiang**,[1,2,3] **Yupeng Ma**,[1,2,3] **and Chunxiang Xu**[1,2,3]

[1]*The Xinjiang Technical Institute of Physics & Chemistry, Urumqi 830011, China*
[2]*Xinjiang Laboratory of Minority Speech and Language Information Processing, Urumqi 830011, China*
[3]*University of Chinese Academy of Sciences, Beijing 100049, China*

Correspondence should be addressed to Tonghai Jiang; jth@ms.xjb.ac.cn

Received 18 May 2018; Accepted 13 September 2018; Published 2 October 2018

Guest Editor: Zhijun Fang

Vehicle detection and classification are very important for analysis of vehicle behavior in intelligent transportation system, urban computing, etc. In this paper, an approach based on convolutional neural networks (CNNs) has been applied for vehicle classification. In order to achieve a more accurate classification, we removed the unrelated background as much as possible based on a trained object detection model. In addition, an unsupervised pretraining approach has been introduced to better initialize CNNs parameters to enhance the classification performance. Through the data enhancement on manual labeled images, we got 2000 labeled images in each category of motorcycle, transporter, passenger, and others, with 1400 samples for training and 600 samples for testing. Then, we got 17395 unlabeled images for layer-wise unsupervised pretraining convolutional layers. A remarkable accuracy of 93.50% is obtained, demonstrating the high classification potential of our approach.

1. Introduction

Vehicle is one of the greatest inventions in human history. The vehicle has become an indispensable part of modern people's life. The use of a huge large number of vehicles can reflect the population's mobility, intimacy, economic, and so on, and the analysis of vehicle behavior is very meaningful for urban development and government decision-making. In order to collect refueling vehicle information, such as license plate, picture, time, location, volume, type, and so on, we have deployed data collecting equipment in many of refueling stations in Xinjiang, which is mainly responsible for safety supervision and analysis of refueling behavior. Till now, many vehicle profile information such as vehicle color and vehicle type is entered into the system by hand; this is inefficient and not uniform. The accurate, various, and volume of data are the key to dig the value of refueling data. Hence, it has become a problem to be solved urgently that how to obtain the vehicle profile information through the vehicle picture automatically. In this paper, we focus on how to get the vehicle type from the picture. This problem is regarded as image classification, which means we should classify the images containing vehicles into the right type by image processing. Due to the environment in which images are taken is quite varied and complex and the impact of irrelevant background, the vehicles in images are very difficult to recognize.

Thanks to the success of deep learning, we present a combination of approaches for vehicle detection and classification based on convolutional neural networks in this paper. To detect the vehicle in the image more efficiently, a successful object detection approach is used to detect the objects in an image, then the target vehicle waiting for entering refueling station is filtered out. Next, we designed a convolutional neural networks which contains 4 convolutional layers, 3 max pooling layers, and 2 full connected layers for vehicle classification. We trained our model on labeled vehicles images dataset. Comparing it with other five state-of-the-art approaches verified our approach achieves the highest accuracy than others. In order to pursue better classification performance, we taken advantage of unsupervised pretraining to better initialize classification model parameters under the circumstance of a shortage of labeled images. The unsupervised pretraining method was implemented based on deconvolution. After pretraining, the convolutional layers

were initialized by the pretrained parameters and trained the model on our labeled images data set; thus, we got a better classification performance than the previous without pretraining.

This paper is organized as follows. The related works are introduced in Section 2. Vehicle detection and classification based on CNNs and a pretraining approach are described in Section 3. In Section 4, the vehicles data set is presented, and we evaluated the presented approaches on our data, and the experimental results and a performance evaluation are given. Finally, Section 5 concludes the paper.

2. Related Works

Existing methods use various types of signal for vehicle detection and classification, including acoustic signal [2–5], radar signal [6, 7], ultrasonic signal [8], infrared thermal signal [9], magnetic signal [10], 3D lidar signal [11] and image/video signal [12–16]. Furthermore, some of the methods can be combined with a variety of signals, such as radar&vision signal [7] and audio&vision signal [17]. Usually, the detection and classification performance is excellent in these methods because of the precise signal data, but there are many hardware devices involved in these methods, resulting in larger deployment cost and even higher failure rate.

The evolution of image processing techniques, together with wide deployment of surveillance cameras, facilitates image-based vehicle detection and classification. Various approaches to image-based vehicle detection and classification have been proposed over the last few years. Kazemi et al. [13] used 3 different kinds of feature extractors, Fourier transform, Wavelet transform, and Curvelet transform, to recognize and classify 5 models of vehicles; k-nearest neighbor is used as classifier. They compare the 3 proposed approaches and find that the Curvelet transform can extract better features. Chen et al. [18] presented a system for vehicle detection, tracking, and classification from roadside closed circuit television (CCTV). First, a Kalman filter tracked a vehicle to enable classification by majority voting over several consecutive frames, then they trained a support vector machine (SVM) using a combination of a vehicle silhouette and intensity-based pyramid histogram of oriented gradient (HOG) features extracted following background subtraction, classifying foreground blobs with majority voting. Wen et al. [19] used Haar-like feature pool on a $32*32$ grayscale image patch to represent a vehicle's appearance and then proposed a rapid incremental learning algorithm of AdaBoost to improve the performance of AdaBoost. Arrospide and Salgado [16] analyzed the individual performance of popular techniques for vehicle verification and found that classifiers based on Gabor and HOG features achieve the best results and outperform principal component analysis (PCA) and other classifiers based on features as symmetry and gradient. Mishra and Banerjee [20] detected vehicle using background, extracted Haar, pyramidal histogram of oriented gradients, shape and scale-invariant feature transform features, designed a multiple kernel classifier based on k-nearest neighbor to divide the vehicles into 4 categories. Tourani and Shahbahrami [21] combined different image/video processing methods including object detection, edge detection, frame differentiation, and Kalman filter to propose a method which resulted in about 95 percent accuracy for classification and about 4 percent error in vehicle detection targets. In these methods, the classification results are very good; however, there are still some problems. First, the image features are limited by the hand-crafted features algorithms to represent rich information. Second, the hand-crafted features algorithms require a lot of calculation, so they are not suitable for real-time applications, especially for embedding in front-end camera devices. Third, most of them are used in fixed scenes and background environments; it is difficult for them to deal with complex environment.

More recently, deep learning has become a hot topic in object detection and object classification area. Wang et al. [22] proposed a novel deep learning based vehicle detection algorithm with 2D deep belief network; the 2D-DBN architecture uses second-order planes instead of first-order vector as input and uses bilinear projection for retaining discriminative information so as to determine the size of the deep architecture which enhances the success rate of vehicle detection. Their algorithm preforms very good in their datasets. He et al. [1] proposed a new efficient vehicle detection and classification approach based on convolutional neural network, the features extracted by this method outperform those generated by traditional approaches. Yi et al. [23] proposed a deep convolution network based on pretrained AlexNet model for deciding whether a certain image patch contains a vehicle or not in Wide Area Motion Imagery (WAMI) imagery analysis. Li et al. [24] presented the 3D range scan data in a 2D point map and used a single 2D end-to-end fully convolutional network to predict the vehicle confidence and the bounding boxes simultaneously, and they got the state-of-the-art performance on the KITTI dataset.

Meantime, object detection and classification based on Convolutional neural networks (CNNs) [25–27] are very successful in the field of computer vision recently. The first work about object detection and classification based on deep learning has been done in 2013; Sermanet et al. [28] present an integrated framework for using deep learning for object detection, localization, and classification; this framework obtains very competitive results for the detection and classifications tasks. Up to now, excellent object detection and classification models based on deep learning include R-CNN [29], Fast R-CNN [30], YOLO [31], Faster R-CNN [32], SSD [33], and R-FCN [34]; these models achieve state-of-the-art results on several data sets. Before the YOLO, many approaches on object detection, for example, R-CNN and Faster R-CNN, repurpose classifiers to perform detection. Instead, YOLO frame object detection is a regression problem to separated bounding boxes and associated class probabilities. The YOLO framework uses a custom network based on the Googlenet architecture, using 8.52 billion operations for a forward pass. However, a more recent improved model called YOLOv2 [35] achieves comparable results on standard tasks like PASCAL VOC and COCO. In YOLOv2 network, it uses a new model, called Darknet-19, and has 19 convolutional layers and 5 maxpooling layers; the model only requires 5.58

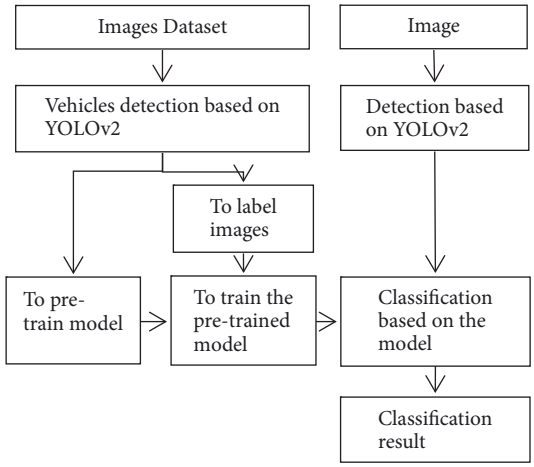

Figure 1: Flowchart of vehicle classification.

billion operations. In conclusion, YOLOv2 is a state-of-the-art detection system, it is better, faster, and stronger than others and applied for object detection tasks in this work.

At last, unsupervised pretraining initializes the model to a point in the parameter space that somehow renders the optimization process more effective, in the sense of achieving a lower minimum of the empirical loss function [36]. Much recent research has been devoted to learning algorithms for deep architectures such as Deep Belief Networks [37, 38] and stacks of autoencoder variants [39]. After vehicle detection, we can easily get a lot of unlabeled images of vehicles and optimize the classification model parameters initialization by unsupervised pretraining.

3. Methodology

In this section, we will present the details of the method based on CNNs for image-based vehicle detection and vehicle classification. This section contains three parts: vehicle detection, vehicle classification, and pretraining approach. The relations between each part and the overall framework of the entire idea is shown in Figure 1.

3.1. Vehicle Detection. Our images taken from static cameras in different refueling station contain front views of vehicles or side views of vehicles at any point. The vehicles in images are very indeterminacy; this makes the vehicle detection more difficult in traditional methods based on hand-crafted features.

The YOLOv2 model is trained on COCO data sets, it can detect 80 common objects in life, such as person, bicycle, car, bus, train, truck, boat, bird, cat, etc., and therefore we can perform vehicle detection based on YOLOv2. In a picture of the vehicle which is waiting for entering the refueling station shown in Figure 2, YOLOv2 can well detect many objects, for example, the security guards, drivers, the vehicle, and queued vehicles, and even vehicles on the side of road. Here, our goal is to pick up the vehicle which is waiting for entering in picture.

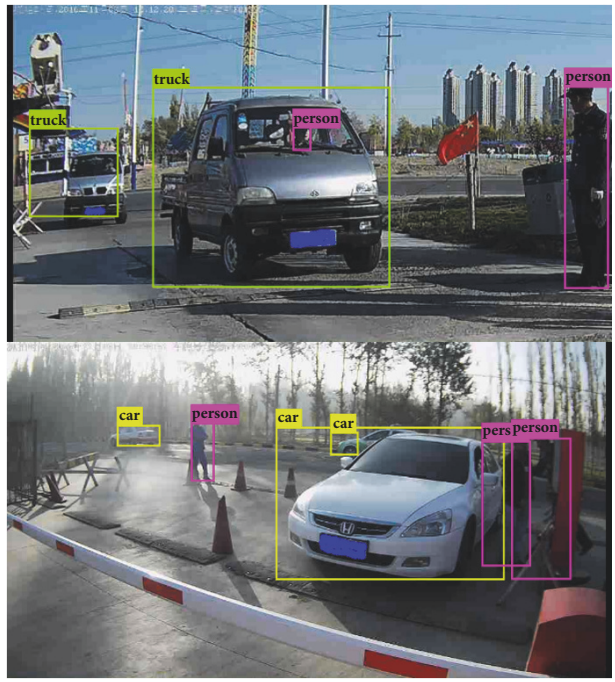

Figure 2: YOLOv2 detection. Top: truck. Down: car.

Although trained YOLOv2 can detect the vehicle and divide it into bicycle, car, motorbike, bus, and truck, it does not meet our classification categories. In order to solve this problem, to fine-tune the YOLOv2 on our data could be a solution, but this method needs amount of manual labeled data and massive computation, it is not a preferable method for us, and then, we presented a rule-based method to detect four categories vehicles more accurately. First, from the YOLOv2 detection results, we select the objects which are very similar to our targets, such as car, bus, truck, and motorbike; second, according to the distance between the vehicle and the camera, the closer the vehicle is, the bigger the target is, we choose the most similar vehicle in picture as the target vehicle entering the refueling station for further vehicle behavior analysis.

3.2. Vehicle Classification. According to the function and size of vehicles, vehicles will be divided into four categories of motorcycle, transporter, passenger, and others. Motorcycle includes motorcycle and motor tricycle; transporter includes truck and container car; passenger includes sedan, hatchback, coupe, van, SUV, and MPV; others include vehicles used in agricultural production and infrastructure, such as tractor and crane, and other types of vehicles. Figure 3 shows the sheared samples in four categories in each column. As we can see, samples images are very different in shape, color, size, and angle of camera, even the samples images in the same category. And Figure 3(b) bottom and Figure 3(c) bottom are not in the same category, but they are very similar, especially on front face, shape, and color, which makes the classification between transporter and passenger more difficult.

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

Figure 3: Sheared vehicle image examples for four categories. (a) Motorcycle; (b) transporter; (c) passenger; (d) others.

Table 1: The structure of C4M3F2.

Layer Type/Activation	Size/Stride	Filters
Convolutional/ReLU	3×3/1	32
Max Pooling	2×2/2	
Convolutional/ReLU	3×3/1	64
Convolutional/ReLU	3×3/1	64
Max Pooling	2×2/2	
Convolutional/ReLU	3×3/1	64
Max Pooling	2×2/2	
Fully Connected/ReLU	1024	
Fully Connected	1024	
Softmax	4	

To solve this difficult problem, we presented a convolutional classification model which is effective and requires little amount of operations. Our model, called C4M3F2, has 4 convolutional layers, 3 max pooling layers, and 2 fully connected layers.

Each convolutional layer contains multiple (32 or 64) 3∗3 kernels, and each kernel represents a filter connected to the outputs of the previous layer. Each max pooling layer contains multiple max pooling with 2×2 filters and stride 2; it effectively reduces the feature dimension and avoids overfitting. For fully connected layers, each layer contains 1024 neurons, each neuron makes prediction from its all input, and it is connected to all the neurons in previous layers. For each sheared vehicle image detected from YOLOv2, it has been resized to 48×48 and then passed into C4M3F2. Eventually, all the features are passed to softmax layer, and what we need to do is just minimizing the cross entropy loss between softmax outputs and the input labels. Table 1 shows the structure of our model C4M3F2.

3.3. Pretraining Approach. With the purpose of achieving a satisfactory classification result, we need more labeled images for training our model, but there is a shortage of labeled images; however, there are plenty of images collected easily, and how to use the plenty of unlabeled images for optimization of our classification model has become the main content in this subsection.

The motivation of this unsupervised pretraining approach is to optimize the parameters in convolutional kernel

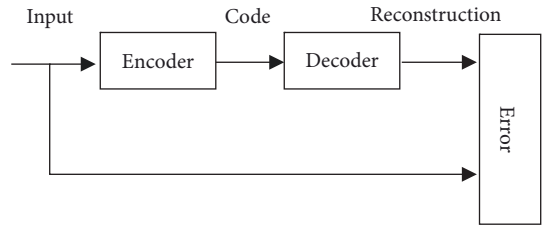

Figure 4: Autoencoder.

parameters. Kernel training in C4M3F2 starts from a random initialization, and we hope that the kernel training procedure can be optimized and accelerated using a good initial value obtained by unsupervised pretraining. In addition, pretraining initializes the model to a point in the parameter space that somehow renders the optimization process more effective, in the sense of achieving a lower minimum of the loss function [36]. Next, we will explain how to greedy layer-wise pretrain the convolution layers and the fully connected layers in the C4M3F2 model. Max pooling layer function is subsampling; it is not included in layer-wise pretraining process.

An autoencoder [40] neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs. It uses a set of recognition weights to convert the input into code and then uses a set of generative weights to convert the code into an approximate reconstruction of the input. Autoencoder must try to reconstruct the input, which aims to minimize the reconstruction error as shown in Figure 4.

According to the aim of autoencoder, our approach for unsupervised pretraining will be explained. In every convolutional layer, convolution can be regarded as the encoder, and deconvolution [41] is taken as the decoder, which is a very unfortunate name and also called transposed convolution. An input image is passed into the encoder, then the output code getting from encoder is passed into the decoder to reconstruct the input image. Here, Euclidean distance, which means the reconstruct error, is used to measure the similarity between the input image and reconstructed image, so the aim of our approach is minimizing the Euclidean paradigm. For pretraining the next layer, first, we should drop the decoder and freeze the weights in the encoder of previous layers and then take the output code of previous layer as the input in this

layer and do the same things as in previous layer. Next how to use transposed convolution to construct and minimize the loss function in one convolutional layer will be described in detail as follows.

The convolution of a feature maps and an image can be defined as

$$y_{ij} = w_i \oplus x_j \tag{1}$$

where $\oplus$ denotes the 2D convolution, $y_{ij} \in R^{r_x \times c_x}$ is the convolution result and the padding is set to keep the input and output dimensions consistent. $w_i \in R^{r_w \times c_w}$ is the ith kernel, and $x_j \in R^{r_x \times c_x}$ denotes the jth training image.

Then, transform the convolution based on circulant matrix in linear system. A circulant matrix is a special kind of Toeplitz matrix where each row vector is rotated one element to the right relative to the preceding row vector. An $n \times n$ circulant matrix C takes the form in

$$C = \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & & c_3 & c_2 \\ \vdots & & \ddots & \vdots & \\ c_{n-2} & c_{n-3} & & c_0 & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{bmatrix} \tag{2}$$

Let $f_j, h_i \in R^{r_e \times c_e}$ be the extension of x_j, w_i, where $r_e = r_x + r_w - 1, c_e = c_x + c_w - 1$. And the method is as follows (3) and (4), where O is zero matrix:

$$f_j = \begin{bmatrix} x_j & O \\ O & O \end{bmatrix} \tag{3}$$

$$h_i = \begin{bmatrix} w_i & O \\ O & O \end{bmatrix} \tag{4}$$

Let $f_j^v \in R^{r_e c_e \times 1}$ be f_j in vectored form, row_a be a row of h_i, and $row_a = [n_0, n_1, n_2, \cdots, n_{c_e-1}]$. To build circulant matrices $H_0, H_1, H_2, \cdots, H_{c_e-1}$ by row_a and by these circulant matrices, a block circulant matrix is defined as shown in formula (5).

$$H = \begin{bmatrix} H_0 & H_{c_e-1} & \cdots & H_2 & H_1 \\ H_1 & H_0 & & H_3 & H_2 \\ \vdots & & \ddots & \vdots & \\ H_{c_e-2} & H_{c_e-3} & & H_0 & H_{c_e-1} \\ H_{c_e-1} & H_{c_e-2} & \cdots & H_1 & H_0 \end{bmatrix} \tag{5}$$

Here, we can transform the convolution into (6).

$$Q = Hf_j^v \tag{6}$$

Q is the vector form of result of convolution calculation and then to reshape Q into $Q' \in R^{r_e \times c_e}$. In this convolution process, the padding is dealing with filling 0, but in actual implementation of our approach, we keep the convolutional input and output dimensions consistent. So we need to prune the extra values to keep the input and output dimensions consistent. So we intercept the matrix Q', then $y_{ij} = Q'[r_w - 1 : r_e - r_w + 1, c_w - 1 : c_e - c_w + 1]$.

To simplify the calculation, we extract the effective rows of H according to the effective row indexes which indicates the position of the elements of y_{ij} in Q and denote these rows by $W_i \in R^{(r_e-2r_w+2)(c_e-2c_w+2) \times 2c_e}$.

Now, $Y_{ij} \in R^{(r_e-2r_w+2)(c_e-2c_w+2) \times 1}$ is vector form of y_{ij}, so the convolution can be rewritten as

$$Y_{ij} = W_i f_j^v \tag{7}$$

There are J training vehicle images and K kernels. Let $X = [f_1^v, f_2^v, \ldots, f_j^v]$, and $W = [W_1; W_2; \ldots; W_K]$.

The convolution can be calculated as

$$Y = \mathrm{W}X \tag{8}$$

And the deconvolution can be calculated

$$X' = W^T Y \tag{9}$$

So X' is the X reconstruction. Then loss function based on Euclidean paradigm is defined in formula (10), being

$$\text{loss}(W, X) = \left\| X - X' \right\|_2 = \sqrt{\sum_{j=1}^{J} \left(X_j - X_j' \right)^2} \tag{10}$$

Then, we used adam optimizer, which is an algorithm for first-order gradient-based optimization of stochastic objective functions based on adaptive estimates of lower-order moments, to solve the minimum optimization problem in formula (10).

After the greedy layer-wise unsupervised pretraining, we initiate the parameters in every convolutional layer with the pretrained values and run the supervised training for classification according to the method in previous subsection.

4. Experiments and Discussions

We evaluated the presented algorithm on our data and compared it with other four state-of-the-art methods.

4.1. Datasets and Experiment Environment. The vehicles images are taken by the static cameras in different refuel stations; after being compressed, they are sent to servers. The quality of images on servers is lower than that selected by random and classified into four categories of motorcycle, transporter, passenger, and others by hand. We got 498 motorcycle images, 1109 transporter images, 1238 passenger images, and 328 other images. Due to the time-consuming and labor-intensive of manual labeling, there is a shortage of labeled images. Image augmentation has been used to enrich the data. Keras, the excellent high level neural network API, provides the ImageDataGenerator for image data preparation and augmentation. Shear range is set to -0.2 to 0.2, zoom range is set to -0.2 to 0.2, rotation range is set to -7 to 7, size is set to 256×256, and the points outside the boundaries are

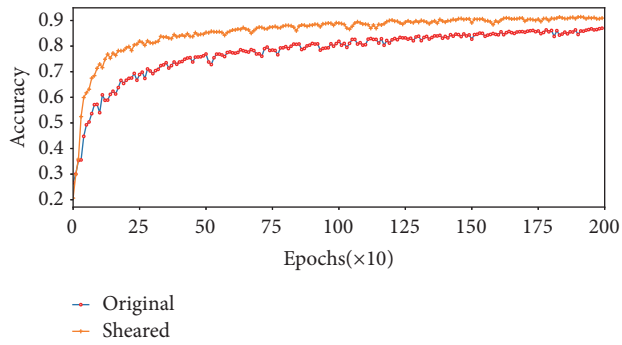

Figure 5: Comparison of training process between original and sheared.

filled according to the nearest mode. After configuration and taking into account the balance of data, we fitted it on our data and got 1400 samples for every categories on the training set and 600 samples for every categories on the testing set to assess our classification model.

For vehicle classification, the CNNs are under Tensorflow framework, the SIFT is under OpenCV (https://opencv.org/), and other feature embedding methods are under scikit-image (http://scikit-image.org/). All the experiments were conducted on a regular notebook PC (2.5-GHz 8-core CPU, 12-G RAM, and Ubuntu 64-bit OS).

4.2. Vehicle Detection Experiment with YOLOv2. For the experiment using original images, the original training set and test set are used for training and testing, for the experiment using sheared images, we used the approach based on trained YOLOv2 to detect the original training set and test set to get the sheared training set and sheared testing set for training and testing.

To verify the importance of vehicle detection for vehicle classification, we designed two groups of vehicle classification experiments, one using original images and the other one using sheared images after vehicle detection, then, the C4M3F2 model is used for vehicle classification experiments.

We initialized the C4M3F2 model by truncated normal distribution, fitted the model on original training set and sheared training set for 2000 epochs, respectively, and recorded the accuracy of our C4M3F2 model on different testing set; the results are shown in Figure 5. As we expected, the sheared images of vehicles more accurately represent the characteristics of vehicles, while pruning more useless information and facilitating the feature extraction and vehicle classification. As we can see in Figure 5, the accuracy of C4M3F2 model using sheared data set is much better than the one using original data set; in the previous training, the characteristics of vehicles were extracted more accurately, so that, the model quickly achieved a better classification results and a stable status.

Finally, the accuracy of C4M3F2 using sheared data set is 91.42%, which is 4.53% higher than the 86.89% of C4M3F2 using original data set. It can be concluded that the results of vehicle classification using sheared data set after vehicle detection based on YOLOv2 can be improved effectively.

Table 2: Accuracy and FPS of different methods.

Method	Accuracy	FPS
HOG+SVM	60.12%	4
DAISY+SVM	69.04%	2
ORB+BoW+SVM	64.07%	7
SIFT+BoW+SVM	74.49%	5
DeCAF[1]	66.20%	13
CNNs	91.42%	800

4.3. Compare Our Approach with Others. There are many other image classification methods. To assess our classification model, we compared our approach with other five methods.

The five methods are based on the image features defined by the scholars in computer image processing. Considering the comprehensive factors, four kinds of image features and a convolutional method are selected, they are histograms of oriented gradient (HOG) [42], DAISY [43], oriented FAST, and rotated BRIEF (ORB)[44], scale-invariant feature transform (SIFT) [45], and DeCAF[1] respectively. These methods are excellent in target object detection in [1, 42–45]. HOG is based on computing and counting the gradient direction histogram of local regions. DAISY is a fast computing local image feature descriptor for dense feature extraction, and it is based on gradient orientation histograms similar to the SIFT descriptor. ORB uses an oriented FAST detection method and the rotated BRIEF descriptors; unlike BRIEF, ORB is comparatively scale and rotation invariant while still employing the very efficient Hamming distance metric for matching. SIFT is the most widely used algorithm of key point detection and description. It takes full advantage of image local information. SIFT feature has a good effect in rotation, scale, and translation, and it is robust to changes in angle of view and illumination; these features are beneficial to the effective expression of targets information. For HOG and DAISY, image features regions are designed; features are computed and sent into SVM classifier to be classified. For ORB and SIFT, they do not have acquisition features regions and specified number of features; we get the image features based on Bag-of-Words (BoW) model by treating image features as words. In the first instance, all features points of all training build the visual vocabulary; in the next place, a feature vector of occurrence counts of the vocabulary is constructed from an image; in the end, the feature vector is sent into SVM classifier to be classified. DeCAF uses five convolutional layers and two fully connected layers to extract features and a SVM to classify the image into the right group [1].

Here, we performed vehicle classification experiments on sheared data set. Table 2 shows the accuracy and FPS of CNNs and other state-of-the-art methods in test; other methods are very slow because they take a lot of time to extract features. It can be observed that the results show the effectiveness of CNNs on vehicle classification problem.

From another point of view, we demonstrated the classification ability of each method by confusion matrix of the

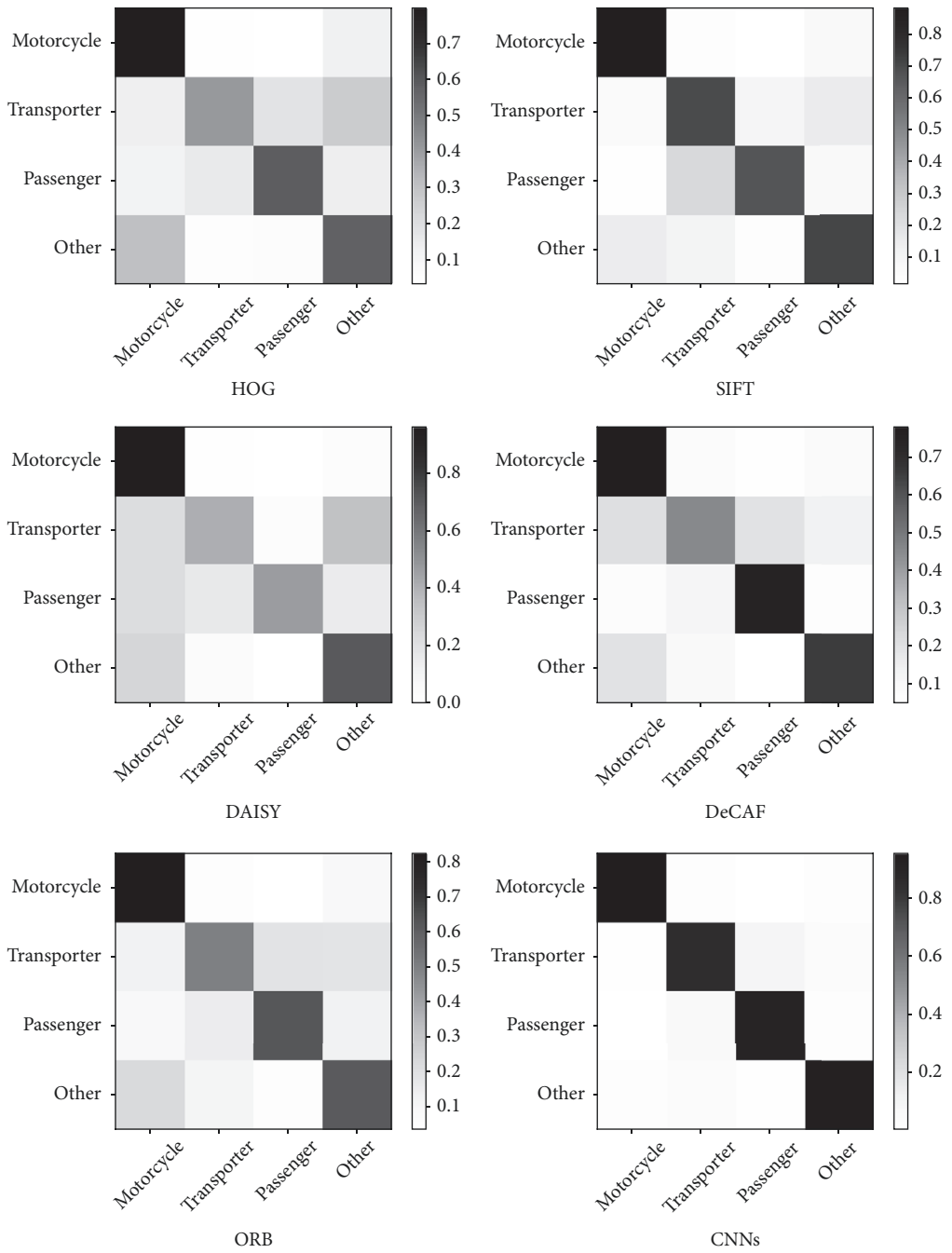

Figure 6: Confusion matrix of classification of different methods.

classification process from the five methods in Figure 6. The main diagonal displays the high recognition accuracy. As shown in Figure 6, the top five comparative methods are better in recognition of motorcycle than other categories. Generally speaking, ORB or SIFT combined with BoW and SVM method is a little better than the other two methods. All taken into account, our CNNs method is the best. But, the performance of CNNs turns out not so satisfactory in view than the confusion of transporter and passenger.

Next, we will focus on the reason of confusion in CNNs. According to the precision, recall, and f1-score of classification in Table 3, it shows that the identification of motorcycle is very good enough, and the identification of transporter and passenger is relatively poor.

As shown in the examples in Figure 7, it can be seen that the wrongly recognized transporter and passenger images include vehicle face information mainly and very little vehicle body information, as far as the main vehicle face is concerned;

Figure 7: Examples of wrongly recognized vehicles images. First row: wrongly recognized as passenger which should be transporter. Second row: wrongly recognized as transporter which should be passenger.

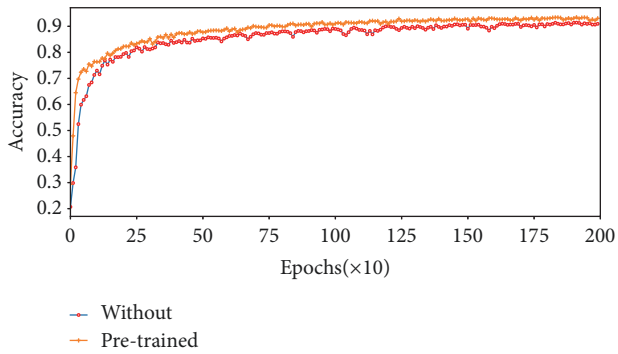

Figure 8: Comparison of training process between pretrained and without pretraining.

Table 3: Classification precision, recall, and f1-score of CNNs.

Type	Precision	Recall	F1-score
Motorcycle	0.97	0.95	0.96
Transporter	0.87	0.85	0.86
Passenger	0.90	0.91	0.90
Other	0.93	0.93	0.93

these vehicles images are so similar in profile that it is still a challenge to recognize same images in Figure 7 manually.

4.4. Pretraining Approach Experiment. We are eager for better performance of our classification model C4M3F2. Here, unsupervised pretraining has been used for optimization of our classification model. 17395 sheared vehicles images are obtained by shearing the unlabeled vehicles images.

We pretrained the parameters of every convolutional layer for 2000 epochs and then supervised training the model on our sheared training set and tested it on our sheared testing set; the results is shown in Figure 8. In the process of training, the conclusion is that the effect is more obvious in the previous epochs, and the overall training process is stable relatively. Ultimately, the accuracy of pretrained CNNs is 93.50%, which is 2.08% higher than the 91.42% of CNNs without pretraining.

Table 4: Classification precision, recall, and f1-score of pretrained CNNs based on sheared dataset.

Type	Precision	Recall	F1-score
Motorcycle	0.99	0.99	0.99
Transporter	0.90	0.99	0.99
Passenger	0.91	0.92	0.92
Other	0.95	0.96	0.95

Table 5: Classification precision, recall, and f1-score of pretrained CNNs based on original dataset.

Type	Precision	Recall	F1-score
Motorcycle	0.99	0.99	0.99
Transporter	0.79	0.82	0.81
Passenger	0.84	0.79	0.81
Other	0.90	0.94	0.92

Table 6: The classification accuracy under different strategies.

	Original	Sheared
CNNs Without pre-training	86.89%	91.42%
CNNs with pre-training	88.29%	93.5%

By analyzing the classification performance of pretrained CNNs, shown in Table 4, we can draw a conclusion that its performance is better than the one of CNNs without pretraining shown in Table 3, especially for the classification of transporter and passenger. In summary, pretrained CNN is more effective in recognizing the vehicles categories, and it is a state-of-the-art approach for vehicle classification.

In the end, to verify the effect of detection in the entire system, we conducted ablation study by pretraining and testing our model on the original dataset which has not been sheared by YOLOv2 and contains a large quantity of irrelevant background. Ultimately, the accuracy of pretrained CNNs on original dataset is 88.29%, which is 5.21% lower than the 93.5% of pretrained CNNs on sheared dataset and even lower than the 91.42% of CNNs without pretraining on sheared dataset; the classification performance is shown in Table 5. And according the classification accuracy in Table 6, we can conclude that this ablation study confirms the essentiality of detection virtually in the whole vehicle classification system.

5. Conclusions

A classification method based on CNNs has been detailed in this paper. To improve the accuracy, we used vehicle detection to removing the unrelated background for facilitating the feature extraction and vehicle classification. Then, an autoencoder-based layer-wise unsupervised pretraining is introduced to improve the CNNs model by enhancing the classification performance. Several state-of-the-art methods have been evaluated on our labeled data set containing four categories of motorcycle, transporter, passenger, and others.

Experimental results have demonstrated that the pretrained CNNs method based on vehicle detection is the most effective for vehicle classification.

In addition, the success of our vehicle classification makes a vehicle color or logo recognition system possible in our refueling behavior analysis; meanwhile, it is a great help to urban computing, intelligent transportation system, etc.

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 is supported by Youth Innovation Promotion Association CAS (2015355). The authors gratefully acknowledge the invaluable contribution of Yupeng Ma and the members of his laboratory during this collaboration.

References

[1] D. He, C. Lang, S. Feng, X. Du, and C. Zhang, "Vehicle detection and classification based on convolutional neural network," in *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, 2015.

[2] J. F. Forren and D. Jaarsma, "Traffic monitoring by tire noise," *Computer Standards & Interfaces*, vol. 20, pp. 466-467, 1999.

[3] J. George, A. Cyril, B. I. Koshy, and L. Mary, "Exploring Sound Signature for Vehicle Detection and Classification Using ANN," *International Journal on Soft Computing*, vol. 4, no. 2, pp. 29–36, 2013.

[4] J. George, L. Mary, and K. S. Riyas, "Vehicle detection and classification from acoustic signal using ANN and KNN," in *Proceedings of the 2013 International Conference on Control Communication and Computing, (ICCC '13)*, pp. 436–439, Thiruvananthapuram, India, 2013.

[5] K. Wang, R. Wang, Y. Feng et al., "Vehicle recognition in acoustic sensor networks via sparse representation," in *Proceedings of the 2014 IEEE International Conference on Multimedia and Expo Workshops, (ICMEW '14)*, pp. 1–4, Chengdu, China, 2014.

[6] A. Duzdar and G. Kompa, "Applications using a low-cost baseband pulsed microwave radar sensor," in *Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference, (IMTC '01)*, vol. 1 of *Rediscovering Measurement in the Age of Informatics*, pp. 239–243, IEEE, Budapest, Hungary, 2001.

[7] H.-T. Kim and B. Song, "Vehicle recognition based on radar and vision sensor fusion for automatic emergency braking," in *Proceedings of the 13th International Conference on Control, Automation and Systems, (ICCAS '13)*, pp. 1342–1346, Gwangju, Republic of Korea, 2013.

[8] Y. Jo and I. Jung, "Analysis of vehicle detection with wsn-based ultrasonic sensors," *Sensors*, vol. 14, no. 8, pp. 14050–14069, 2014.

[9] Y. Iwasaki, M. Misumi, and T. Nakamiya, "Robust vehicle detection under various environments to realize road traffic flow surveillance using an infrared thermal camera," *The Scientific World Journal*, vol. 2015, Article ID 947272, 2015.

[10] J. Lan, Y. Xiang, L. Wang, and Y. Shi, "Vehicle detection and classification by measuring and processing magnetic signal," *Measurement*, vol. 44, no. 1, pp. 174–180, 2011.

[11] B. Li, T. Zhang, and T. Xia, "Vehicle Detection from 3D Lidar Using Fully Convolutional Network," https://arxiv.org/abs/1608.07916, 2016.

[12] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, 2003.

[13] F. M. Kazemi, S. Samadi, H. R. Poorreza, and M.-R. Akbarzadeh-T, "Vehicle recognition based on fourier, wavelet and curvelet transforms - A comparative study," in *Proceedings of the 4th International Conference on Information Technology-New Generations, ITNG '07*, pp. 939-940, Las Vegas, Nev, USA, 2007.

[14] J. Y. Ng and Y. H. Tay, "Image-based Vehicle Classification System," https://arxiv.org/abs/1204.2114, 2012.

[15] R. A. Hadi, G. Sulong, and L. E. George, "Vehicle Detection and Tracking Techniques: A Concise Review," *Signal & Image Processing: An International Journal*, vol. 5, no. 1, pp. 1–12, 2014.

[16] J. Arróspide and L. Salgado, "A study of feature combination for vehicle detection based on image processing," *The Scientific World Journal*, vol. 2014, Article ID 196251, 13 pages, 2014.

[17] P. Piyush, R. Rajan, L. Mary, and B. I. Koshy, "Vehicle detection and classification using audio-visual cues," in *Proceedings of the 3rd International Conference on Signal Processing and Integrated Networks, (SPIN '16)*, pp. 726–730, Noida, India, 2016.

[18] Z. Chen, T. Ellis, and S. A. Velastin, "Vehicle detection, tracking and classification in urban traffic," in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems, ITSC '12*, pp. 951–956, Anchorage, Alaska, USA, 2012.

[19] X. Wen, L. Shao, Y. Xue, and W. Fang, "A rapid learning algorithm for vehicle classification," *Information Sciences*, vol. 295, pp. 395–406, 2015.

[20] P. Mishra and B. Banerjee, "Multiple Kernel based KNN Classifiers for Vehicle Classification," *International Journal of Computer Applications*, vol. 71, no. 6, pp. 1–7, 2013.

[21] A. Tourani and A. Shahbahrami, "Vehicle counting method based on digital image processing algorithms," in *Proceedings of the 2nd International Conference on Pattern Recognition and Image Analysis, IPRIA '15*, pp. 1–6, Rasht, Iran, 2015.

[22] H. Wang, Y. Cai, and L. Chen, "A vehicle detection algorithm based on deep belief network," *The Scientific World Journal*, vol. 2014, Article ID 647380, 7 pages, 2014.

[23] M. Yi, F. Yang, E. Blashch et al., "Vehicle Classification in WAMI Imagery using Deep Network," in *Proceedings of the SPIE 9838: Sensors and Systems for Space Applicatioins IX*, 2016.

[24] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D lidar using fully convolutional network," in *Proceedings of the Robotics: Science and Systems*, 2016.

[25] Y. Lecun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Neural Information Processing Systems*, pp. 1097–1105, 2012.

[28] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," https://arxiv.org/abs/1312.6229, 2013.

[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, Columbus, Ohio, USA, 2014.

[30] R. Girshick, "Fast R-CNN," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1440–1448, Santiago, Chile, 2015.

[31] 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 '16)*, pp. 779–788, Las Vegas, Nev, USA, 2016.

[32] 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.

[33] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the Computer Vision – ECCV 2016*, vol. 9905 of *Lecture Notes in Computer Science*, pp. 21–37, 2016.

[34] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," https://arxiv.org/abs/1605.06409, 2016.

[35] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, (CVPR '17)*, pp. 6517–6525, Honolulu, Hawaii, USA, 2017.

[36] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[37] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[38] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Neural Information Processing Systems*, pp. 153–160, 2007.

[39] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction," in *Proceedings of the Artificial Neural Networks and Machine Learning - ICANN 2011*, vol. 6791 of *Lecture Notes in Computer Science*, pp. 52–59, Springer, Berlin, Heidelberg, Germany, 2011.

[40] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and helmholtz free energy," *Neural Information Processing Systems*, pp. 3–10, 1994.

[41] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR' 10)*, pp. 2528–2535, San Francisco, Calif, USA, 2010.

[42] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, 2005.

[43] E. Tola, V. Lepetit, and P. Fua, "DAISY: an efficient dense descriptor applied to wide-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, 2010.

[44] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 2564–2571, Barcelona, Spain, 2011.

[45] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 2, pp. 1150–1157, Kerkyra, Greece, 1999.