J Inf Process Syst, Vol.17, No.2, pp.411~425, April 2021
https://doi.org/10.3745/JIPS.01.0073

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Vehicle Image Recognition Using Deep Convolution Neural Network and Compressed Dictionary Learning

Yanyan Zhou*

Abstract

In this paper, a vehicle recognition algorithm based on deep convolutional neural network and compression dictionary is proposed. Firstly, the network structure of fine vehicle recognition based on convolutional neural network is introduced. Then, a vehicle recognition system based on multi-scale pyramid convolutional neural network is constructed. The contribution of different networks to the recognition results is adjusted by the adaptive fusion method that adjusts the network according to the recognition accuracy of a single network. The proportion of output in the network output of the entire multiscale network. Then, the compressed dictionary learning and the data dimension reduction are carried out using the effective block structure method combined with very sparse random projection matrix, which solves the computational complexity caused by high-dimensional features and shortens the dictionary learning time. Finally, the sparse representation classification method is used to realize vehicle type recognition. The experimental results show that the detection effect of the proposed algorithm is stable in sunny, cloudy and rainy weather, and it has strong adaptability to typical application scenarios such as occlusion and blurring, with an average recognition rate of more than 95%.

Keywords

Adaptive Fusion, Compressed Dictionary Learning, Deep Convolutional Neural Network, High-Dimensional Features, Vehicle Type Recognition

1. Introduction

Vehicle type identification is an important part of the intelligent transportation system. The rational use of vehicle type identification system not only brings convenience to the traffic management department, but also has important strategic significance for solving traffic congestion and planning urban traffic system [1-3]. At present, vehicle recognition based on computer vision mainly focuses on vehicle feature extraction and classifier construction. In feature extraction, the gray level co-occurrence matrix can describe the spatial information of the relative positions of different gray level pixels in the vehicle images, but this feature cannot fully grasp the characteristics of local gray level graphics. For larger parts, the extracted texture features cannot be used for vehicle type recognition [4,5]. Gabor wavelet can well describe both spatial and frequency domains of the image signal and has good class representation ability. However, the Gabor feature dimension is high and the calculation is heavy, so it is not suitable for real-time vehicle recognition [6]. In the aspect of classifier construction, the support vector machine and the neural network have good classification effects.

Manuscript received September 4, 2020; first revision October 21, 2020; accepted November 8, 2020.

Corresponding Author: Yanyan Zhou (yyz07099@163.com)
* School of Mathematics and Computer, Tongling University, Tongling, Anhui, China (yyz07099@163.com)

In order to improve the real-time and accuracy of vehicle type recognition in complex traffic scenarios, a vehicle type recognition method based on information fusion is proposed in [7]. The main innovation of this method was to match the data collected by multiple sensors to form the feature waveform of bicycle fusion. Lin et al. [8] proposed an unsupervised feature learning method based on K-feature for vehicle recognition in infrared images. The unsupervised feature learning algorithm of K-feature was used to learn visual dictionary from a large number of unlabeled samples, extract features and generate unsupervised feature learning algorithm of K-feature to suppress false alarm and improve the learning accuracy. The authors of [9] established an improved in-depth learning method for vehicle type recognition based on surveillance images and proposed a system based on convolutional neural network (CNN) and transfer learning, which was implemented by tags from network data.

In the traditional vehicle recognition system, manual feature extraction is often used. The quality of feature selection depends on manual selection. The CNN can learn features adaptively driven by the training data, which is more representative than the features extracted manually. Moreover, the CNN has good resistance to affine deformation such as displacement and scaling, and can effectively overcome the influence of some external conditions on vehicle appearance. Therefore, this paper uses a CNN to learn features. In this paper, a precise vehicle type recognition method based on multi-scale pyramid-connected CNN is proposed. The pyramid-connected CNN is used to extract local and global features of low-level and high-level for precise vehicle type recognition. Furthermore, the multi-scale pyramid-connected CNN is used to increase the richness of features.

2. Feature Fusion Strategy Based on Convolutional Neural Network

2.1 Block Diagram of Vehicle Type Recognition Based on Convolutional Neural Network

The overall block diagram of fine vehicle recognition based on the CNN is shown in Fig. 1. The whole process of vehicle recognition consists of two parts, one is the training part and the other is the testing part. In the training part, the gray and normalized vehicle face sample image is used as the input of the CNN. The feature of the vehicle image is extracted by the CNN. Then, the compressed dictionary is learned. The effective block structure method and the very sparse random projection matrix are used to reduce the dimension of the data, which solves the problem of high dimension. The computational complexity caused by the features shortens the time of dictionary learning. Finally, the sparse representation classification method is used to realize vehicle type recognition [10].

The training process of the CNN is a supervised learning process, which includes the forward and the back-propagation stages. In the forward propagation stage, the training samples are input into the network to calculate the actual output of the network. The error between the actual output and the ideal output is calculated in the back-propagation stage. The parameters are adjusted and the network is optimized by the back-propagation algorithm and the stochastic gradient descent method, so that the network is adjusted in the direction of global optimum [11,12]. In the initial training stage of the CNN, the error rate of network recognition decreases rapidly, but with the increase of the training time, the error rate decreases slowly. When the training reaches a certain number of iterations, the accuracy remains unchanged, and it is difficult to increase the accuracy even if the training iterations are increased.

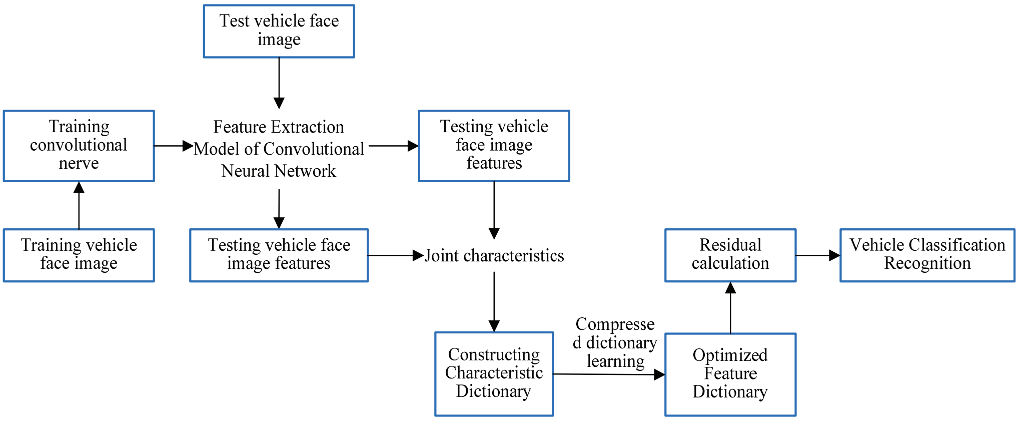

Test vehicle face image
Training convolutional nerve
Feature Extraction Model of Convolutional Neural Network
Testing vehicle face image features
Training vehicle face image
Testing vehicle face image features
Joint characteristics
Residual calculation
Vehicle Classification Recognition
Compressed dictionary learning
Constructing Characteristic Dictionary
Optimized Feature Dictionary

Fig. 1. General block diagram of fine vehicle type recognition based on convolution neural network.

In order to extract more abundant features suitable for fine vehicle recognition, the network structure of the CNN is divided into two parts, extracting local information of low level and global information of high level. The concrete structure of the pyramid CNN used in this paper is shown in Fig. 2. In both components of the network, there are alternately connected convolutional layer and pooling layer. The convolutional layer is used to extract the features and make them invariant in rotation and translation. The pooling layer is used to reduce the computational complexity and make the features more robust to small deformation [13,14].

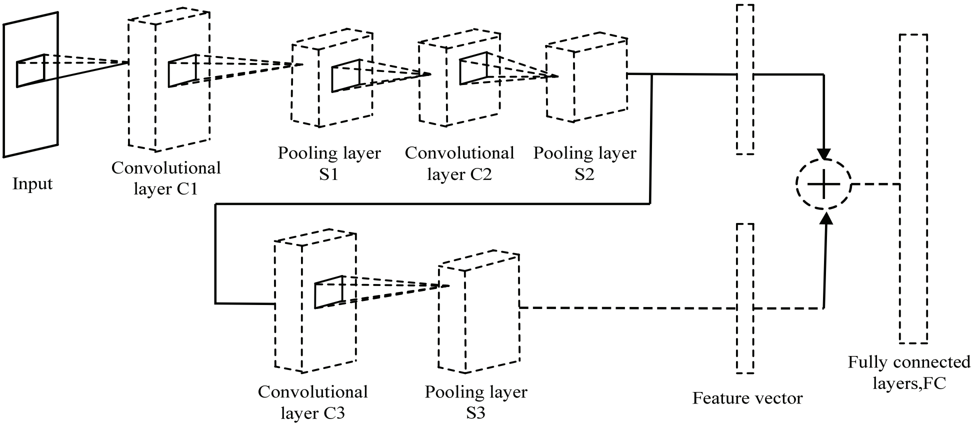


Fig. 2. The concrete structure of the pyramidal convolution neural network.

Because the number of neurons output from the pooling layer S3 is small, it is easy to cause the bottleneck of information transmission if only full connection is made with the pooling layer S3. Adding the information of the S2 layer can reduce the information loss which may be caused by the convolution process of the C3 layer. Therefore, the connection mode of the pooling layer S2 in Fig. 2 has changed. On the one hand, the feature map of the S2 layer is vectorized to obtain the information loss. On the other hand, the feature map in the S2 layer is convoluted with the convolution kernel to obtain the convolutional layer C3, which then passes through the pooling layer S3 to obtain another part of the feature vector.

These two parts of eigenvectors contain the local information of the lower level and the global information of the higher level in the vehicle face image. The final eigenvector is composed of these two parts. The final eigenvector is compressed and learnt in the dictionary. The effective block structure method and the very sparse random projection matrix are used to reduce the dimension of the data. The problem of computational complexity caused by the high-dimensional features is solved, and the dictionary learning time is shortened. Finally, the sparse representation classification method is used to realize vehicle type recognition.

2.2 Feature Fusion Strategy

After the region extraction network is used to obtain the region of interest (recommendation box), the corresponding feature areas on the feature map of each convolutional layer are calculated according to the obtained recommendation box to fuse the features. The receptive field is one of the most important concepts in the convolutional neural networks. It is defined as a region in the input space corresponding to a specific convolutional feature.

Through the receptive field, the corresponding feature areas of the region of interest (ROI) on the feature map can be calculated, and the corresponding feature areas in the feature map of different channels behind the same convolutional layer are the same. Firstly, the corresponding feature areas on the feature map generated by the first layer of the convolution are calculated along the width of the ROI of the vehicle. Formula (1) calculates the number of features corresponding to the ROI on the feature map along the wide direction. Then the distance between the features on the feature map is calculated using the formula (2), and finally, the location of the first feature along the wide direction on the feature map is calculated using the formula (3).

$$n_{wout} = \left[\frac{n_{win}}{S} \right] \tag{1}$$

$$j_{out} = j_{in} * S \tag{2}$$

$$S_{wout} = S_{win} + \left[\frac{k-1}{2} \right] * j_{out} \tag{3}$$

where, n_{win} represents the width of the ROI; j_{in} represents the distance between the pixels in the ROI, with a value of 1; j_{out} represents the distance between the features on the feature map; S represents the step of the convolution core sliding, k represents the size of the convolution core; S_{win} represents the starting value of the ROI along the wide direction in the input image and S_{wout} represents its corresponding features. The method of calculating the corresponding feature region along the high direction of the ROI is the same as the method for calculating the width along the wide direction. First, using formula (4), the corresponding feature of the ROI of the vehicle along the high direction in the convolutional layer is calculated. Then, the formula (5) is used to calculate the position of the first feature point along the high direction on the feature map.

$$n_{hout} = \left[\frac{n_{hin}}{S} \right] \tag{4}$$

$$S_{hout} = S_{hin} + \left[\frac{k-1}{2} \right] * j_{out} \tag{5}$$

In the formula, n_{hin} represents the height of the ROI, n_{hout} represents the height of the ROI corresponding to the feature area on the feature map, S_{hin} represents the starting value of the ROI along the high direction in the input image, and S_{hout} represents the starting value of the corresponding feature area along the high direction on the feature map. Therefore, (S_{wout}, S_{hout}) is the coordinate of the first feature point in the ROI, and $(S_{wout} + n_{wout}, S_{hout} + n_{hout})$ is the coordinate of the last feature point. When calculating the corresponding feature areas of ROI on other convolutional layers, the upper feature map can be used as an input image, and the formula mentioned above can be used recursively to calculate.

The resolution of feature maps output from different convolutional layers is different, and the size of the feature areas corresponding to the ROI of vehicles is also different. Therefore, it is necessary to unify the feature areas obtained from different convolutional layers to the same size. In this paper, ROI pooling is used to achieve the goal. Firstly, along the channel direction of each feature region, each feature region is evenly divided into M×N blocks [15]. As mentioned in the previous paper, the fused features will be connected to the fifth layer convolutional layer of the deep residual network. Although the convolutional layer can accept any size of features, the subsequent design of full connection layer and softmax will not be affected by global average pooling. However, in order to avoid large differences with the pre-training parameters, this paper divides the feature regions into 62×37 blocks. Then each feature block is maximized and the maximum value in the feature block is used as the output feature of the feature block. After ROI pooling, the feature regions extracted from different convolutional layers have the same resolution.

Through ROI pooling, the feature regions extracted from different layers have the same resolution, but the magnitude of features in different layers is different because of convolution calculation at different levels. Therefore, it is necessary to standardize the features of each layer in order to connect the features of different layers. This paper chooses the L2 standardization in the channel, which is calculated by the formula (6). In formula (6), x_i represents the feature points in the feature graph, n represents the number of feature points in the feature graph, and y_i represents the value of the corresponding feature points after L2 standardization.

$$y_i = \frac{x_i}{\sum_{i=0}^{n} \sqrt{x_i^2}} \tag{6}$$

After ROI pooling and data standardization (L2), the feature regions extracted from each layer have the same resolution and operable quantities. The fused features can be obtained by connecting the feature maps extracted from each convolutional layer along the channel direction. In order to reduce the computational complexity and ensure the similarity between the original residual network and ROI, this paper adds a scale layer and reduces the dimension of the fused feature through a 1×1 convolutional layer, so that it is consistent with the output dimension of the fourth convolutional layer of ResNetlol. At this time, the features obtained will be applied in vehicle detection network [16,17].

2.3 Adaptive Fusion Classification Results

The fusion method of multiple single network recognition results directly affects the recognition effect of the entire network. If the fusion method is selected properly, the overall classification accuracy of the network is significantly improved than that of the single network [18]. This paper uses an adaptive fusion method to better fuse the recognition results of each hopping layer CNNs network and adjust the contribution of the recognition results of a single network to the final results. The adaptive fusion method adjusts the proportion of the output of the network in the entire multi-scale network according to the recognition accuracy of a single network.

After training the network, the test sample is used as the input of the network. Each network in the entire multi-scale pyramidal convolutional neural network classifies the test data and obtains multiple output results and classification accuracy.

If the classification accuracy of the *i*-th clinopyramid CNNs is Ac_i, the number of tested images is N, and the number of correctly recognized images is N_r, then the calculation formula of Ac_i is as follows:

$$Ac_i = \frac{N_r}{N} \tag{7}$$

The output of the network i is denoted as $O(y)_i$, where $O(y)_i$ is shown in (8). Each column in $O(y)_i$ represents a sample, and P_{ij} represents the probability that the classifier classifies sample j as class i. r represents the number of training samples in batch training.

$$O(y)_i = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1r} \\ P_{21} & P_{22} & \cdots & P_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ P_{k1} & P_{k2} & \cdots & P_{kr} \end{bmatrix} \tag{8}$$

The output of CNNs with multiple scales is adaptively fused to obtain the total output of the whole network, which is denoted as $O(y)$. The calculation formula of adaptive fusion is as follows:

$$O(y) = \sum_{i=1}^{n} \frac{Ac_i}{\sum_{i=1}^{n} Ac_i} \tag{9}$$

Denominator $\sum_{i=1}^{n} Ac_i$ is used to normalize data. The final classification result is the category label corresponding to the maximum value of each column in output $O(y)$.

3. Compressed Dictionary Learning

After feature extraction and association, all joint features are constructed into sparse representation dictionaries. Since the dimension of the sparse representation dictionary constructed is too high, using the traditional K-SVD dictionary learning algorithm for dictionary training will inevitably lead to complex computation problems [14,19]. In this paper, the block structure is added to the dictionary learning process. Sample data sets are divided into data blocks of the same size, and very sparse random projection matrix is used to reduce the dimension of data blocks and reduce the computational load of dictionary learning to enhance the effectiveness of dictionary learning [20]. The specific data processing process is as follows.

(1) Sample data is processed in blocks. Assuming that the original training data sample set is $\boldsymbol{X}$, it is divided into L data blocks of the same size, namely $\boldsymbol{X} = \left[\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \cdots, \boldsymbol{X}^{(l)}, \cdots, \boldsymbol{X}^{(L)}\right]$, where $\boldsymbol{X}^{(l)}$ is the l data block of the training data sample set.

(2) Using very sparse random projection matrices $\boldsymbol{R}_l \in \boldsymbol{R}^{p\times m}$, $m<p$, and the l data block of the training sample set, the inner product is obtained.

$$\boldsymbol{Y}^{(l)} = \boldsymbol{R}_l^T \boldsymbol{X}^{(l)}, 1 \le l \le L \tag{10}$$

In the formula, $\boldsymbol{Y}^{(l)}$ is the l-th data block of $\boldsymbol{X}^{(l)}$ after dimensionality reduction of the very sparse random projection matrix. Each independent random variable r_{ij} in $\left\{R_l\right\}_{l=1}^{L}$ satisfies the distribution:

$$r_{ij} \sim \begin{bmatrix} -1 & 0 & +1 \\ \frac{1}{2s} & 1-\frac{1}{s} & \frac{1}{2s} \end{bmatrix} \tag{11}$$

In formula, i and j are rows and columns of independent random variables, and parameter s controls the sparsity of random projection matrix. On average, each column in $\left\{R_l\right\}_{l=1}^{L}$ contains p non-zero elements.

3.1 Sparse Coding

In the sparse coding stage, fixed dictionary $\boldsymbol{D}$, effective block structure is adopted and Batch-OMP method is used on each sample data block to obtain the optimal sparse coefficient matrix $\boldsymbol{C}^{(l)}$ of training sample $\boldsymbol{Y}^{(l)}$ on the dictionary $\boldsymbol{D}$. The sparse coding process can be summarized as solving the following optimization problems:

$$\begin{cases} \min\limits_{C \in R^{K\times n}} \sum\limits_{l=1}^{L} \left\| \boldsymbol{Y}^{(l)} - \boldsymbol{R}_l^T \boldsymbol{D}\boldsymbol{C}^{(l)} \right\|_F^2 \\ s.t.\ \forall i, \left\| \boldsymbol{c}_i^{(l)} \right\|_0 \le T \end{cases} \tag{12}$$

In the formula, $\boldsymbol{c}_i^{(l)}$ is the i-th training sample parameter in the l-th block coefficient matrix $\boldsymbol{C}^{(l)}$, $\boldsymbol{D}=\left[d_1, d_2, \cdots, d_k\right]$ is the training dictionary, $\boldsymbol{C}=\left[C_1, C_2, \cdots, C_n\right]$ is the sparse coefficient matrix, and $\left\|c_i^{(l)}\right\|_0$ is the number of non-zero values in the control sparse coefficient vector.

3.2 Dictionary Update

The purpose of dictionary updating is to update each column of atoms in the dictionary and its corresponding sparse coefficients. In the dictionary updating stage, assuming that the atom d_j in the dictionary $\boldsymbol{D}$ is fixed and the atom d_k in the dictionary $\boldsymbol{D}$ is updated iteratively, repeating $k+1$ times until $k = K$, so that the dictionary and the sparse coefficient are updated synchronously. The process of dictionary updating can be summarized as solving the following optimization problems:

$$\begin{aligned}\sum_{l=1}^{L}\left\|\boldsymbol{Y}^{(l)}-\boldsymbol{R}_l^T\boldsymbol{D}\boldsymbol{C}^{(l)}\right\|_F^2 &= \sum_{l=1}^{L}\sum_{i=1}^{nl}\left\|y_i^{(l)}-\boldsymbol{R}_l^T\sum_{j=1}^{K}\boldsymbol{c}_{i,k}^{(l)}d_j\right\|_2^2 \\ &= \sum_{l=1}^{L}\sum_{i=1}^{nl}\left\|y_i^{(l)}-\boldsymbol{R}_l^T\sum_{j\neq k}^{K}\boldsymbol{c}_{i,k}^{(l)}d_j-\boldsymbol{c}_{i,k}^{(l)}\boldsymbol{R}_l^T d_k\right\|_2^2 \\ &= \sum_{l=1}^{L}\sum_{i=1}^{nl}\left\|\boldsymbol{e}_{i,k}^{(l)}-\boldsymbol{c}_{i,k}^{(l)}\boldsymbol{R}_l^T d_k\right\|_2^2\end{aligned} \tag{13}$$

where $\boldsymbol{c}_{i,k}^{(l)}$ is the sparse coefficient vector of column k in $\boldsymbol{c}_i^{(l)} \in \boldsymbol{R}^K$, d_k is the fixed dictionary atom of column k, and $\boldsymbol{e}_{i,k}^{(l)} = y_i^{(l)} - \boldsymbol{R}_l^T\sum_{j\neq k}^{K} c_{i,k}^{(l)}d_j \in \boldsymbol{R}^m$ is the representation error of column $y_i^{(l)}$ after removing the dictionary atom of column k. In formula (13), the objective function is a quadratic function problem about dictionary atom d_k. The derivative of the objective function about d_k is equal to zero to solve its optimal value. First of all, let $\tau_k^{(l)} = \left\{i \middle| 1 \le i \le n_l, \boldsymbol{c}_{i,k}^{(l)} \neq 0\right\}$ be available.

$$G_k d_k = b_k \tag{14}$$

Formula $G_k \triangleq \sum_{l=1}^{L} s_k^{(l)}\boldsymbol{R}_l\boldsymbol{R}_l^T$; $b_k \triangleq \sum_{l=1}^{L}\sum_{i\in\tau_k^{(l)}} c_{i,k}^{l}\boldsymbol{R}_l e_{i,k}^{(l)}$; $s_k^{(l)}$ is the sum of squares of all coefficients related to the lexicographic atoms of column k in the l data sample block, and $s_k^{(l)} \triangleq \sum_{l=1}^{L}\sum_{i\in\tau_k^{(l)}}\left(c_{i,k}^{l}\right)^2$.

In order to solve G_k effectively, all random projection matrices in the series are $\boldsymbol{R}$, $\boldsymbol{R} \triangleq [R_1, R_2, \cdots, R_L] \in \boldsymbol{R}^{p\times(mL)}$, and the diagonal matrix $\boldsymbol{S}$ is defined as:

$$\boldsymbol{S}_k \triangleq diag\left(\left[s_k^{(1)},\cdots,s_k^{(l)},\cdots,s_k^{(L)},\cdots,s_k^{(L)}\right]\right) \tag{15}$$

where $diag(z)$ denotes the diagonal square matrix, G_k and $G_k = \boldsymbol{R}\boldsymbol{S}_k\boldsymbol{R}^T$ are obtained, and the updated k column dictionary atom d_k is obtained by solving $G_k d_k = b_k$. Finally, the optimal sparse parameters $c_{i,k}^{(l)}$ and $c_{i,k}^{(l)} = \left\langle \boldsymbol{e}_{i,k}^{(l)}, \boldsymbol{R}_l^T d_k \right\rangle / \left\|\boldsymbol{R}_l^T d_k\right\|_2^2, \forall i \in \tau_k^{(l)}$ are obtained by least square method. After the renewal of one row of atoms, continue to update the next row of atoms until all the atoms in the dictionary have been updated. Generate the optimized new dictionary $\boldsymbol{D}' = \left[d_1', d_2', \cdots, d_K'\right]$.

4. Vehicle Type Recognition Based on Sparse Representation

Sparse representation classification uses training samples sparsely and linearly to represent the test samples, and classifies them according to the minimization of the class linear reconstruction error. In this paper, a sparse representation classification model is established by using the compressed feature dictionary, and the vehicle type recognition is realized by calculating the minimum reconstruction error of the target to be measured in the dictionary. Using sparse representation to classify vehicle types, the steps are as follows.

(1) Assume that the feature dictionary obtained after learning the compressed dictionary is $\boldsymbol{D}'=\left[d_1',d_2',\cdots,d_K'\right]\in\boldsymbol{R}^{p\times m}$, Here d_i' is the feature matrix of training samples of class i, $i=1,2,\cdots,K$, p is the dimension of data samples, and n is the total number of training samples.

(2) According to the principle of linear space, assuming that any sample $y\in\boldsymbol{R}^p$ from a training sample to be identified can be represented by a linear combination of the training samples, the optimal solution can be solved by using the l_1-norm minimization problem when the value of the sparse coefficient $\boldsymbol{x}$ is sparse enough. Its objective function is as follows:

$$\hat{\boldsymbol{x}}=\arg\min_{x}\left\{\left\|\boldsymbol{y}-\boldsymbol{D}'\boldsymbol{x}\right\|_2^2+\lambda\left\|\boldsymbol{x}_1\right\|\right\} \tag{16}$$

Where λ is a scalar constant.

(3) The feature function δ_i is defined to retain the elements related to class i in sparse coefficient $\hat{\boldsymbol{x}}$, and the rest is set to 0. The feature function is used to approximate the samples to be identified, and the reconstruction errors are compared with those of the samples to be identified. The samples to be identified are classified into the corresponding categories of the minimum reconstruction errors:

$$identity\left(y\right)=\arg\min_{i}\left\{\boldsymbol{e}_i\right\} \tag{17}$$

Formula $\boldsymbol{e}_i=\left\|\boldsymbol{y}-d_i'\delta_i\left(\hat{\boldsymbol{x}}\right)\right\|_2$ denotes the reconstruction error between the identified sample and the class i sample, while $\hat{\boldsymbol{x}}=\left[\hat{\boldsymbol{x}}_1;\hat{\boldsymbol{x}}_2;\cdots;\hat{\boldsymbol{x}}_K\right]$ and $\hat{\boldsymbol{x}}_i$ denote the coefficient vectors associated with the class i sample.

5. Experimental Results and Analysis

In order to verify the effectiveness of the proposed algorithm, the following experiments are carried out. An intelligent traffic camera is set up on the road overpass to collect vehicle images. The collected images are rotated (+3 degrees) to simulate the change of vehicle angle. At the same time, the number of samples is increased. The collected images are grayed and the vehicle and the background images are manually segmented. There are four types of buses, trucks, minibuses and cars. Each type of vehicle takes 150 images as samples. After rotating, each type of vehicle includes 450 samples, and the size of each image is uniform 100×100 pixels. The input image size is 64×64, the convolution core size of C1 and C2 layers is 5×5, and C1 layer contains 12 feature maps of 60×60 size. Through the pooling area of 2×2, the feature maps of S1 layer with 12 sizes of 30×30 are obtained. The C2 layer contains 24 feature maps of 26×26 size. Similarly, through the pooling layer S2, the number of feature maps remains unchanged, and the size of feature maps is reduced to 13×13. Thirty-six convolution cores of 4×4 size are convoluted with S2 to obtain the convolutional layer C3 that contains 36 feature maps of 10×10 size. After pooling layer S3, 36 feature maps of 5×5 are finally obtained. Finally, the feature maps from the S2 and S3 layers are vectorized to form the final feature vectors. In the experiment, the impulse coefficient is 0.95.

5.1 Comparisons of Dictionary Learning Algorithms

To verify the efficiency and accuracy of the compressed dictionary learning algorithm, the proposed algorithm is compared with the classical K-SVD dictionary learning algorithm. The running time and the average accuracy of the algorithm are compared under different training samples. The experimental results are shown in Table 1 and Fig. 3. It can be seen from Table 1 that the running time of the two algorithms is very close when the training samples are small, because when the training data is small, the data dimension has little influence on the computing time and the time required for both algorithms is similar.

Table 1. Dictionary training time (unit: second)

Experimental methods	Number of training samples							
	200	400	600	800	1,000	1,200	1,400	1,600
K-SVD	3.02	5.21	7.25	9.87	14.56	18.68	25.32	31.25
Proposed method	2.95	3.45	3.98	4.58	4.96	5.24	5.65	6.02

However, with the increase of the training samples, the time spent on dictionary training by using the compressed dictionary learning algorithm is longer than that of the classical K-SVD dictionary. The learning algorithm decreases because when the training samples increase, the influence of data dimension on the computing time increases. After partitioning the sample data set, the algorithm uses Batch-OMP algorithm to solve the sparse coefficient, which is faster and can add a very sparse random projection matrix to the dictionary learning. It can effectively reduce the dimension of the sample data, significantly reduce the amount of calculation in the training process, and shorten the running time of the algorithm. It can be seen from Fig. 3 that the average accuracy of the proposed algorithm is slightly lower than that of the classical K-SVD dictionary learning algorithm when the number of training samples is small. This is because part of the information will be lost when the dimensionality is reduced. Therefore, the accuracy of the proposed algorithm is slightly lower than that of the classical K-SVD algorithm when the number of training samples is small. However, as the number of training samples increases, the accuracy of the proposed algorithm becomes better than that of the classical K-SVD algorithm.

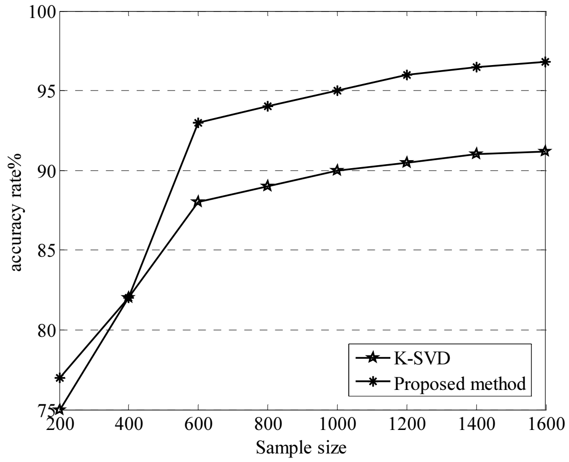


Fig. 3. Comparison of recognition rate between the proposed algorithm and K-SVD algorithm.

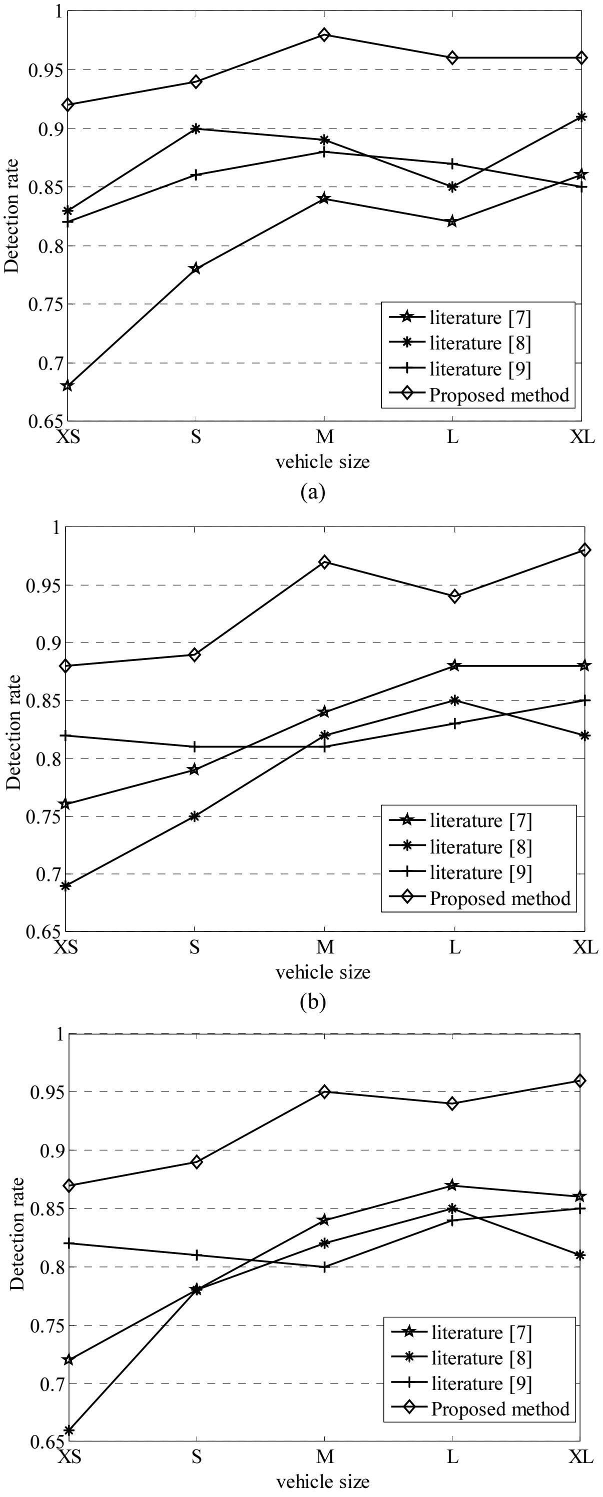
5.2 Comparisons with Other Vehicle Recognition Methods

In addition to comparing the overall performance of the algorithm, this paper also analyzed the performance of each detection algorithm to detect vehicle targets of different sizes. In the actual driving environment, the number of vehicle targets in different frames is different. In order to maintain the stability of the algorithm, according to the method proposed by Hoiem et al. [21], this paper divides the vehicle targets into five categories based on size: XS, S, M, L and XL, and calculates the average vehicle target size and the aspect ratio. Five reference frames are extracted for each feature point on the output feature map of the fourth convolutional layer according to the mean of the size and the aspect ratio of the five types of vehicle targets. Then, according to the detection results of the previous frame, the basic reference frame is modified to some extent. For each vehicle target detected in the previous frame, it is first divided into different categories according to its size, and the corresponding feature areas on the feature map are calculated. Then, according to its size and aspect ratio, nine reference frames are extracted from the feature interval of 3×3 around the center of the feature area, and the basic reference frames at the response feature points are covered.

In this section, the detection results of the proposed vehicle detection algorithm in different traffic scenarios are analyzed to verify the robustness of the proposed algorithm. The driving scene can be divided into three basic conditions according to the weather conditions: sunny day, cloudy day and rainy day. Different road conditions and light changes may exist in different weather conditions without further details. Because the video data in this paper are collected by the monocular camera, only two kinds of vehicle detection algorithms based on prior information and traditional machine learning are selected for lateral comparison. The actual comparison algorithms include the algorithm in this paper, the algorithms in [7], [8], and [9]. The experimental results are shown in Fig. 4.

In sunny weather, the image quality is relatively high, and the features of each object in the driving scene are relatively obvious, which is conducive to the extraction of vehicle features. In some scenarios, there are fewer vehicle targets and the features are relatively obvious. All the algorithms have achieved good performance. Among them, the average recognition rate of the proposed algorithm is more than 95%. In some scenarios, there are changes in light, such as slow changes in light conditions over time, and sudden changes in light when the surrounding buildings are shielded from the sun. Different illumination conditions will affect the performance of detection algorithm, especially under special illumination conditions, vehicle targets will have similar colors with road or surrounding environment, and the detection performance based on horizontal edge algorithm will be significantly reduced. The experimental results also show that the expressive ability of the convolution feature is relatively stronger. This paper proposes a vehicle detection algorithm that utilizes both shallow features with perfect details and deep features with strong expressive ability. Therefore, the recognition effect of the proposed algorithm for small vehicle targets is better than the other three algorithms, and the introduction of compressed dictionary learning is also improved. The detection performance of the algorithm for vehicle targets with similar background color is studied.

The cloudy weather is similar to the sunny weather, but the image is relatively darker, which interferes with the feature extraction. Compared with the sunny day, the performance of each detection algorithm has a certain degree of decline, especially in two algorithms of [8] and [9], but the algorithm using convolution characteristics with strong expression ability is relatively less affected.



(a)

(b)

(c)

Fig. 4. Comparison of different detection algorithms in different weather condition: (a) clear/sunny day, (b) cloudy day, and (c) rainy day.

Rainy days are the most difficult to deal with. The video frames collected during the rain are as dark as the cloudy days. At the same time, the raindrops will have a certain impact on the detection of vehicle targets, especially the smaller vehicle targets. When using the wipers, the images will become more ambiguous and the detection of vehicle targets will become more difficult. Because many situations need to be dealt with on rainy days, feature extraction is more difficult. Compared with sunny weather, the detection performance of each algorithm has a large decline, especially the vehicle detection algorithm based on horizontal edges. The comparison of the false detection rate of the two algorithms using the convolution feature is relatively small.

The proposed algorithm achieves a good balance between detection performance and computational efficiency. The test of robustness subdivides the data set. The experimental results show that the detection effect of the proposed algorithm is stable in sunny, cloudy and rainy weather conditions, and it is more adaptable to typical application scenarios such as occlusion and blurring.

6. Conclusion

In order to overcome the computational time-consuming problem caused by the lack of identification information for a single feature and the high dimension of sample data, a vehicle recognition algorithm based on deep CNN and compression dictionary is proposed in this paper. The CNN is used to learn the features. The traditional CNN recognition system only uses the top-level information to classify. This top-level information cannot well represent the input image and cannot meet the need of fine identification of vehicle brand and model. Therefore, this paper proposes a fine vehicle type recognition method based on multi-scale pyramid connected CNN. Jump-connected CNN is used to extract the low-level local features and the high-level global features suitable for fine vehicle recognition, and multi-scale jump-connected CNN is further used to increase the richness of features.

In this paper, the block structure is added to the dictionary learning process. Sample data sets are divided into data blocks of the same size, and very sparse random projection matrix is used to reduce the dimension of data blocks and reduce the computational load of dictionary learning in order to enhance the effectiveness of dictionary learning. Moreover, since the labeling of each image in the database used is done by the author himself, it is unavoidable that there are sorting errors and inconsistent calibration standards due to the enormous workload.

References

[1] K. Yoneda, A. Kuramoto, and N. Suganuma, “Convolutional neural network based vehicle turn signal recognition,” in *Proceedings of 2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Okinawa, Japan, 2017, pp. 204-205.

[2] D. Chowdhury, S. Mandal, D. Das, S. Banerjee, S. Shome, and D. Choudhary, “An adaptive technique for computer vision based vehicles license plate detection system,” in *Proceedings of 2019 International Conference on Opto-Electronics and Applied Optics (Optronix)*, Kolkata, India, 2019, pp. 1-6.

[3] K. F. Hussain and G. S. Moussa, “On-road vehicle classification based on random neural network and bag-of-visual words,” *Probability in the Engineering and Informational Sciences*, vol. 30, no. 3, pp. 403-412, 2016.

[4] J. Yang, T. Liu, B. Jiang, H. Song, and W. Lu, “3D panoramic virtual reality video quality assessment based on 3D convolutional neural networks,” *IEEE Access*, vol. 6, pp. 38669-38682, 2018.

[5] U. Muhammad, W. Wang, and A. Hadid, “Feature fusion with deep supervision for remote-sensing image scene classification,” in *Proceedings of 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, Volos, Greece, 2018, pp. 249-253.

[6] Y. Song, G. Yang, H. Xie, D. Zhang, and X. Sun, “Residual domain dictionary learning for compressed sensing video recovery,” *Multimedia Tools and Applications*, vol. 76, no. 7, pp. 10083-10096, 2017.

[7] F. Li and Z. Lv, “Reliable vehicle type recognition based on information fusion in multiple sensor networks,” *Computer Networks*, vol. 117, pp. 76-84, 2017.

[8] J. Lin, Y. Tan, H. Xia, and J. Tian, “Infrared vehicle recognition using unsupervised feature learning based on K-feature,” in *Proceedings of SPIE 10608: MIPPR 2017: Automatic Target Recognition and Navigation*. Bellingham, WA: International Society for Optics and Photonics, 2018. https://doi.org/10.1117/12.2288698

[9] J. Wang, H. Zheng, Y. Huang, and X. Ding, “Vehicle type recognition in surveillance images from labeled web-nature data using deep transfer learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2913-2922, 2017.

[10] Y. Chen, W. Zhu, D. Yao, and L. Zhang, “Vehicle type classification based on convolutional neural network,” in *Proceedings of 2017 Chinese Automation Congress (CAC)*, Jinan, China, 2017, pp. 1898-1901.

[11] B. Hicham, A. Ahmed, and M. Mohammed, “Vehicle type classification using convolutional neural network,” in *Proceedings of 2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, Marrakech, Morocco, 2018, pp. 313-316.

[12] E. Zheng, D. Ji, E. Dunn, and J. M. Frahm, “Self-expressive dictionary learning for dynamic 3D reconstruction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 9, pp. 2223-2237, 2017.

[13] J. Li, Y. Song, Z. Zhu, and J. Zhao, “Highly undersampled MR image reconstruction using an improved dual-dictionary learning method with self-adaptive dictionaries,” *Medical & Biological Engineering & Computing*, vol. 55, no. 5, pp. 807-822, 2017.

[14] Y. Zhao, L. Meng, X. Wang, and F. Li, “Research on performance classification of modified asphalt mixture based on clustering algorithm,” *Journal of Building Materials*, vol. 17, no. 3, pp. 437-445, 2014.

[15] P. Sharma and P. Bajaj, “Accuracy comparison of vehicle classification system using interval type-2 fuzzy inference system,” in *Proceedings of 2010 3rd International Conference on Emerging Trends in Engineering and Technology*, Goa, India, 2010, pp. 85-90.

[16] M. Lin and X. Zhao, “Application research of neural network in vehicle target recognition and classification,” in *Proceedings of 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, Changsha, China, 2019, pp. 5-8.

[17] Y. T. Xu, S. T. Zhao, D. Jiang, and J. F. Ren, “The role of improved k-means clustering algorithm in the motion parameters determination of Breaker's moving contact,” *Advanced Materials Research*, vol. 960, pp. 905-909, 2014.

[18] L. Peng, M. Peng, B. Liao, Q. Xiao, W. Liu, G. Huang, and K. Li, “A novel information fusion strategy based on a regularized framework for identifying disease-related microRNAs,” *RSC Advances*, vol. 7, no. 70, pp. 44447-44455, 2017.

[19] Z. Yibo, L. Qi, and H. Peifeng, "Vehicle type classification system for expressway based on improved convolutional neural network," in *Proceedings of 2020 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, Chengdu, China, 2020, pp. 78-82.

[20] Z. Dong and J. Jia, "Vehicle type classification using distributions of structural and appearance-based features," in *Proceedings of 2013 IEEE International Conference on Image Processing*, Melbourne, Australia, 2013, pp. 4321-4324.

[21] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV)*, Beijing, China, 2005, pp. 654-661.

Yanyan Zhou **https://orcid.org/0000-0001-5360-9176**

She was born in 1978 in Zongyang, Anhui Province, China. She has got her master's degree, and she is currently a lecturer in Tongling University. Her main research interests include image processing and artificial intelligence