Wang Q, Ding YD. A novel fine-grained method for vehicle type recognition based on the locally enhanced PCANet neural network. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(2): 335–350 Mar. 2018. DOI 10.1007/s11390-018-1822-7

A Novel Fine-Grained Method for Vehicle Type Recognition Based on the Locally Enhanced PCANet Neural Network

Qian Wang[1,2,3] and You-Dong Ding[3,4], *Senior Member, CCF*

[1]*School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China*
[2]*Information Center, Criminal Investigation Department of Shanghai Public Security Bureau, Shanghai 200083, China*
[3]*Shanghai Engineering Research Center of Motion Picture Special Effects, Shanghai 200072, China*
[4]*Shanghai Film Academy, Shanghai University, Shanghai 200072, China*

E-mail: cathyiii@hotmail.com; ydding@shu.edu.cn

Received October 11, 2016; revised November 20, 2017.

Abstract In this paper, we propose a locally enhanced PCANet neural network for fine-grained classification of vehicles. The proposed method adopts the PCANet unsupervised network with a smaller number of layers and simple parameters compared with the majority of state-of-the-art machine learning methods. It simplifies calculation steps and manual labeling, and enables vehicle types to be recognized without time-consuming training. Experimental results show that compared with the traditional pattern recognition methods and the multi-layer CNN methods, the proposed method achieves optimal balance in terms of varying scales of sample libraries, angle deviations, and training speed. It also indicates that introducing appropriate local features that have different scales from the general feature is very instrumental in improving recognition rate. The 7-angle in 180° (12-angle in 360°) classification modeling scheme is proven to be an effective approach, which can solve the problem of suffering decrease in recognition rate due to angle deviations, and add the recognition accuracy in practice.

Keywords fine-grained classification, PCANet, local enhancement, vehicle type recognition

1 Introduction

In recent years, many human and vehicle-based image recognition techniques have been used in the video investigation applications in terms of face recognition[1-5], human and vehicle highlight detection[6], license recognition, detection of individual or crowd behaviors, calculation of crowd density and vehicle flows, and the recognition of traffic violations. Benefiting from these techniques, the researchers began to perform in-depth studies on a lot of applications in order to further distinguish genders, ages, and behaviors of passengers, determine whether the vehicles are large-, medium- or small-sized, and check whether the vehicles are motorcycles, electric assisted bicycles, traditional bicycles, or disabled cars. Even the brand[7] or color of the vehicles is expected to be recognized for subsequent fine-grained classification. The purpose of vehicle type recognition in this paper is to identify a vehicle's model, not just its brand (manufacturer). For example, given a vehicle, we attempt to not only tell you that it is an Audi car, but also check whether it is A6 or A8.

Due to the technical requirements on the attitude angle of the target for face, body and vehicle recognition, many monitoring resources are abandoned, underscoring the need for full-angle and multi-attitude vehicle recognition. Compared with the needs from general car manufactures and consumers, those from public security management in evidence determination hope to achieve a higher recognition accuracy. Considering the timeliness of investigation, case analysis is usually expected to be performed as soon as possible. In extreme cases like temporary monitoring in the wild, the scheme that can operate in real time without high-performance hardware is also required. To sum up, we need a full-angle vehicle recognition technique that can operate

Regular Paper

©2018 Springer Science + Business Media, LLC & Science Press, China

accurately and efficiently without heavy resource consumption.

In the past three to four years, a broad variety of methods were proposed to classify objects that have a high similarity to one another, such as cats, dogs, birds, flowers, and trees. This kind of problems is called the fine-grained categorization[2-3,8-10] problems. The vehicle type recognition, as one of the most prominent features of vehicles which could be identified by computer vision, is also included in this categorization, but little attention is paid to it. Since 2013, the research team led by Li at Stanford University[1] and the research team led by Tang at the Chinese University of Hong Kong[2] have been engaged in vehicle type recognition. The two teams constructed the Stanford Cars[11] and the Compcar[12] datasets, respectively, offering benefits to subsequent researches. But there is little in-depth study on this issue. The fine-grained recognition algorithm represents the state-of-the-art one[12-15], to the best of our knowledge. Existing methods have the following limitations. 1) Pre-processing of images like normalization and labeling involves heavy workloads and makes these methods infeasible for the case with a large number of samples[1-2,13]. 2) The use of too many interactive schemes[3,9] makes it impossible to perform autonomous recognition quickly, easily with little intervention. And some tasks like the work in [13-14, 16] cannot be fulfilled by ordinary personnel without relevant expertise. 3) They are mostly effective for a particular scenario but not robust to other various environments[7,17]. 4) Moreover, due to their inefficiency and high-level computational complexity, these methods are unsuitable for subsequent applications and cannot be implanted into real-time systems[3-5,18-19].

The paper focuses on the fine-grained recognition of different vehicle types, and proposes a novel fine-grained recognition algorithm based on the locally enhanced PCANet neural network[20]. The contribution of this paper is as follows.

1) We construct a professional dataset with 1 797 vehicle types (90% vehicle types registered in Shanghai) and over 2.5 million vehicle images in total. These images are carefully and standardly classified in a multi-angle and multi-property manner. It surpasses the existing databases in terms of the number, type, standardization and classification grained.

2) We pioneer the use of PCANet for vehicle type recognition and propose a 7-angle classification modeling scheme by sampling data at every 30° angle which is the minimum interval in industry applications. Meanwhile, we implement a local feature selection scheme based on simple labeling, and a fine-grained vehicle type recognition algorithm that combines the weighted local features with the general feature.

3) We propose an unsupervised network learning algorithm, resulting in greatly reduced pre-processing workloads at the early stage. Compared with other neural network algorithms, the proposed method adopts fewer network layers and substantially reduces the number of training samples needed for network convergence; optimal trade-off has been achieved for fewer computational loads, greater calculating efficiency, and higher recognition rate compared with popular convolutional neural networks (CNNs). Results based on the new dataset constructed in this paper and other open datasets demonstrate the superiority of the proposed algorithm, and also indicate that the proposed algorithm is particularly effective for the medium and large datasets and the dataset of extremely similar vehicle types. Hence, the proposed algorithm is well suited for practical applications.

2 Related Work

The vehicles originated from the period 1970∼1980. Vehicle types represent different products among the car manufacturers at different time. The vehicles are unique in terms of production technology, quality, appearance and assembly. In this context, the vehicle type is an important aspect of vehicle classification. Due to the rapid advance of the modern manufacturing industry and the heightened awareness of customization, the types of private cars have exploded in recent 10 years. In the past, there is no special dataset that focuses on vehicles only, because the vehicle was only collected as an ordinary sort of objects in some datasets, such as the ImageNet[3] which includes the "car&elevator car" category as well as the category "automobile" in CIFAR-10 and CIFAR-100 dataset[4]. In 2013, Stanford Cars[11] (Scar, for short) collected 16 697 images about 197 mainstream vehicle types made since 1990. In 2015, Compcar[12] (Comp, for short) collected

[1] http://vision.stanford.edu/index.html, Nov. 2017.

[2] http://mmlab.ie.cuhk.edu.hk/, Nov. 2017.

[3] http://image-net.org/synset?wnid=n02960352, Nov. 2017.

[4] http://www.cs.toronto.edu/˜kriz/cifar.html, Nov. 2017.

141 727 images about 1 687 local vehicle types made in the past 10 years. Regarding the source of data, most images of Scar and Comp are derived from the Internet. Comp has additional 5 000 images obtained through video surveillance. Regarding the high-similarity types and the variants of the same type (the same types made in different years), Scar only includes 512 small-scale images of 10 high-similarity sorts (BMW-10), and most of the remaining images are large-scale distinguishable ones (Car-197). Comp incorporates many variants of the same vehicle types and regards them as different sorts. Obviously, it adds difficulty to image recognition. Note that both datasets attach great importance to the viewpoint of vehicles in the belief that the viewpoint has a large influence on vehicle recognition. Comp classifies the images at each angle and computes the number of images accordingly. In addition to the two famous open datasets described above, researchers built datasets according to their own needs. But most of these datasets only contain a small number of samples[14-16,19] of few categories[14-16,21] and some of them are too special[14,16]. Note that the Car-333 dataset in [22] contains 157 023 non-labeled training images and 7 840 test images of 333 vehicle types. It is the largest one of existing medium-sized datasets. But the dataset that we construct using the monitoring images contains 2.5 million labeled images in 1 797 types. Our dataset is superior to Car-333 in terms of the number of images and data standardization.

The increase of the number of vehicle types and the development of open databases facilitate the studies on vehicle type recognition and some studies have already been done to solve this problem. The authors in [13] adopted the state-of-the-art SPM (Spatial Pyramid Matching) and BB (BubbleBank) models. They concluded that extracting local features from the small-scale datasets is more effective than extracting general feature and local features, and the reverse is true for large-scale datasets. The part-based DPM (Deformable Parts Model) is adopted in [14-15]. Deep learning theories, especially the currently popular CNN algorithms, are used in [12, 16-17, 21-23] to classify vehicle types. For example, a semi-supervised CNN method is used in [21] to classify six types of vehicles. The authors of [22] formulated the fine-grained model and the hyper-class recognition model, and then improved the recognition accuracy by mining the relationship between the two models. In [17], different levels of relevance between images are studied to propose a multi-task learning framework. In order to address the problem of recognizing vehicle types under different viewpoints, the authors in [13-16, 24] proposed to construct 3D features by applying the 2D features to the 3D geometric model. The developers of Comp[12] divided all images into five viewpoints using the annotator and then improved the recognition accuracy by establishing viewpoint-wise models, and they concluded that the CNN-based full-angle model is more accurate than the angle-wise models. The authors of [16] proposed to extract the unpacked version of the vehicles, vehicle front, side and roof to constitute the rasterized bounding boxes, which are then combined with the encoded viewpoints before being input to CNN. In [15], the vehicle images are divided into eight angles, with 45 degrees as a sector to improve the recognition accuracy by classifying the non-labeled images according to the viewpoint. But over 12% vehicle images are misrecognized to wrong angle divisions, resulting in a reduction in recognition accuracy in some cases. A hyper class label is used in [22], which provides fresh insights into image labeling. The authors of [13] and the authors of [12, 14] spent a lot of energy on manual landmarks based on Scar and self-developed dataset respectively. It is reported in [15] that non-labeled images cause considerable reduction in recognition accuracy.

To sum up, the vehicle type recognition dataset was developed very lately, only a small number of vehicle types are included, the viewpoints (or angles) differ greatly, and it is difficult to label the images. In addition, the vehicle type recognition methods mostly rely on the fine-grained recognition theory. Due to what mentioned above, the literature on vehicle type recognition is much less than that on the recognition of other objects. Hence, review of research on this issue is incorporated into the review of research on fine-grained recognition methods and frameworks. Existing solutions to the fine-grained classification problem can be categorized into three types.

1) *Classification Based on the Construction of Middle-Level Local Features.* SIFT (scale invariant feature transform)[25], SURF (speedup robust features) and HOG (histogram of oriented gradient)[26] are typical examples of local features. Specifically, HOG is robust, computationally efficient and invariant to illumination variation and slight deviation. Currently, it is still used to detect pedestrians and construct low-level features. The work in [8] focuses on the search for semantically meaningful features. The method in [9] cannot be implemented without user interaction. The authors of [10] proposed a non-parametric component

transferring method and reduced the labeling time by simply labeling the components in the datasets. The method proposed in [1] is very effective in image classification and retrieval, as it uses the Fisher vector (FV) as the low-level features to generate the middle-level features through BoW (bag of words). This method yields insights into structuring of video images. The authors in [2] proposed the local POOF features, which provide an effective approach to distinguish the location and size of the same features between different classes. This proposal is very inspirational but it needs sufficient local features to represent the whole object, resulting in increased labeling workload.

2) *General Classification Based on Deep Learning.* In [3], an algorithm is proposed, which uses the learning similarity metrics to combine perceptual and visual information for categorization in an integrated manner. But the choice of features in this method mostly relies on the color of the same type of animal hair. Convolution neural network and recursive convolution neural network are used very successfully in [18] to recognize faces. Results indicate that by adjusting a large number of parameters, the method can recognize faces more accurately than the traditional methods. DeepID[4] and DeepID$_2$[5] perform face recognition using the convolution neural network. Specifically, DeepID$_2$ introduces the ideas of metrics learning and the fully connected layer of the network incorporates features on different scales which are important to final results. All of the similar deep learning networks[4-5,18], however, require hours or even days of training as well as a large sample library, in addition to the complicated process of parameter tuning.

3) *Construction of Features by Combining Locality with Generality.* This type of algorithms is mostly seen in real-world applications. While flexibly adopting various classic and popular methods, this type of algorithms adapt the advantages and disadvantages of various methods according to application requirements.

PCANet[20] is a novel deep learning framework proposed by Chan *et al.* in 2015. It is very simple in structure (only two layers in current applications) and highly extendable, needs few parameters, and allows the parameters to be tuned without much expertise. Moreover, it is rotation-invariant during feature extraction and very robust against the noise. Combining the advantages of traditional pattern recognition and neural network methods, results on many databases indicate that its recognition performance is comparable to that of CNN with 10-plus or tens of layers, without the use of a library of millions of samples as in the deep neural network. Since the PCANet was proposed, it has been used in fine-grained recognition field by several researchers, such as road sign recognition[27], grass seed classification[28], histopathological image classification[29], loop closure detection[30], and even image quality evaluation[31]. This algorithm is comparable to CNN in terms of complexity, resource consumption and recognition accuracy. The number of samples and labels this algorithm requires is affordable in the vehicle type recognition applications. Therefore, it is our chief choice in this paper and its advantages will be expounded in Section 3.

3 Proposed Vehicle Type Recognition

3.1 General Framework

Our proposed method for fine-grained classification of vehicle types ensures recognition rate without the compromise of efficiency and simplicity. The main idea of the proposed method is to achieve fine-grained classification of vehicle types by using PCANet, which combines the weighted small-scale local features and the large-scale general feature. Finally, the vehicle types are classified through the Softmax regression function classifier.

3.2 Feature Extraction

In this paper, we will take the 0° modeling as an example to illustrate the entire process of feature extraction. But when we establish the model for each angle, we will choose different parts of the vehicle for local feature extraction and then fuse the extracted features into the final feature vector. Details of the scheme will be described below.

3.2.1 *Choice and Extraction of General Feature*

Extraction of general feature of vehicle types is illustrated in Fig.1. The entire PCANet network can be classified into five steps described as follows.

1) *Input.* Let I denote the set of front vehicle images with a size of 600 480 after being normalized. Also let I_{train} denote 80% of the images and I_{test} denote the remaining 20%. We firstly define I_{train} as the training dataset and then convert it into matrices $\{\boldsymbol{I}_i\}_1^N$.

2) *PCA Filter* 1. We partition the images into blocks according to the pixel with a size of $k_1 \times k_2$. After block sampling, all sampled blocks are cascaded

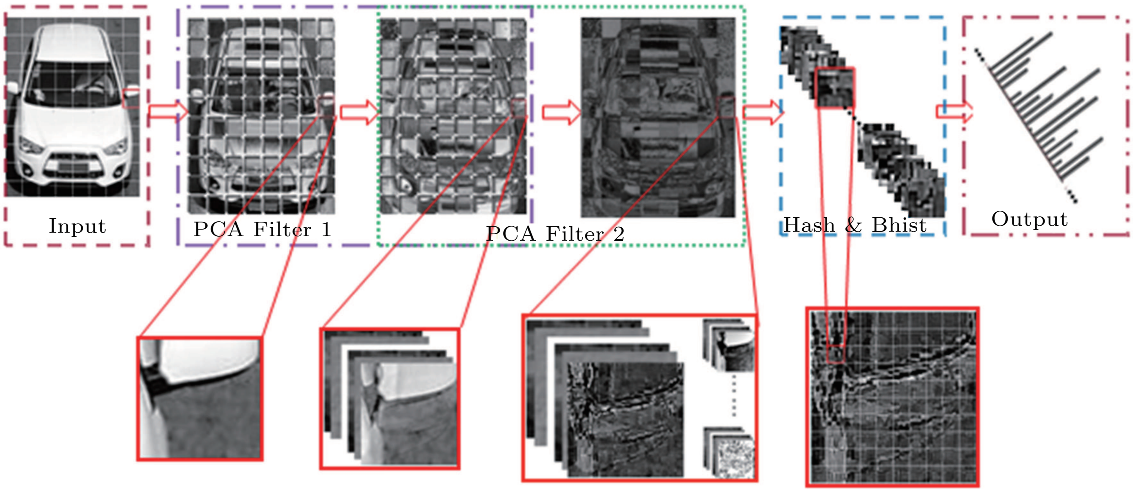


Fig.1. Extraction of general feature of vehicle types.

to represent each image $x_{i,j}$ as:

$$x_{i,j} = x_{i,1}x_{i,2}x_{i,3}\cdots x_{i,\tilde{m}\tilde{n}} \in \mathbb{R}^{k_1k_2}, \quad (1)$$

where $x_{i,j}$ in (1) denotes the j-th block in the $\boldsymbol{I}_i$ image, $\mathbb{R}^{k_1k_2}$ represents the set of all blocks, and $\tilde{m}\tilde{n}$ denotes the number of blocks after the image is partitioned into $\tilde{m}\times\tilde{n}$ blocks. Subtracting the average $\bar{x}_i$ from each column to get N after-block-treated matrices to generate $\boldsymbol{X}$:

$$\boldsymbol{X} = (\bar{x}_{i,1}, \bar{x}_{i,2}, \bar{x}_{i,3}, \cdots, \bar{x}_{i,N}) \in \mathbb{R}^{k_1k_2\times N\tilde{m}\tilde{n}}. \quad (2)$$

Let L_s denote the s-th filter. In order to minimize the reconstruction error by searching for multiple orthonormal matrices, we acquire:

$$\min_{\boldsymbol{V}\in\mathbb{R}^{k_1k_2\times L_1}} \|\boldsymbol{X} - \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{X}\|_F^2$$
$$\text{s.t. } \boldsymbol{V}^{\mathrm{T}}\boldsymbol{V} = \boldsymbol{I}_{L_1}. \quad (3)$$

In (3), we determine L_s by converting it to the simple PCA, where $\boldsymbol{I}_{L_1}$ is the identity matrix of size $L_1 \times L_1$. That is, the sample image is reconstructed by extracting the feature vectors of the first L_1 feature values of $\boldsymbol{X}$ calculated in (2). In this way, the output of PCA filter 1 $\{\boldsymbol{W}_l^1\}$ is obtained as:

$$\boldsymbol{W}_l^1 = mat_{k_1,k_2}(q_l\left(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\right)) \in \mathbb{R}^{k_1k_2},$$
$$l = 1, 2, \cdots L_1, \quad (4)$$

where $mat_{k_1,k_2}(q_l\left(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\right))$ is a function that maps $\mathbb{R}^{k_1k_2}$ to $\boldsymbol{W}_l^1$, and $q_l(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})$ in (4) is the s-th principal eigenvectors feature vector of $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$. Then, by computing the convolution of each image $\boldsymbol{I}_i$ with $\boldsymbol{W}_l^1$, we can obtain L_1 which denotes the number of $\boldsymbol{I}_i^l$, and then make this set of images to be used as the input to PCA filter 2.

3) *PCA Filter* 2. The algorithm for this layer is almost the same as PCA filter1. The only difference is that the input is changed to $\{\boldsymbol{I}_i^l\}$, inputting $L_1 \times N$ image matrices. The feature vectors corresponding to the first L_2 feature values of $\boldsymbol{Y}$ are extracted to reconstruct the sample images, yielding $\{\boldsymbol{W}_l^2\}$ as the output of PCA filter 2:

$$\boldsymbol{W}_l^2 = mat_{k_1,k_2}\left(q_l\left(\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\right)\right) \in \mathbb{R}^{k_1k_2},\ l = 1, 2, \cdots, L_2, \quad (5)$$

where $\boldsymbol{Y}$ in (5) denotes the NL_1 training sample matrices obtained after modularization, cascading and mean subtraction of $\{\boldsymbol{I}_i^l\}$.

$$\boldsymbol{Y}_i^l = (\bar{y}_{i,l,1}, \bar{y}_{i,l,2}, \cdots, \bar{y}_{i,l,N}) \in \mathbb{R}^{k_1k_2\times N\tilde{m}\tilde{n}},$$

which can also be expressed as:

$$\boldsymbol{Y} = \left(\boldsymbol{Y}^1, \boldsymbol{Y}^2, \cdots, \boldsymbol{Y}^{L_1}\right) \in \mathbb{R}^{k_1k_2\times L_1N\tilde{m}\tilde{n}}. \quad (6)$$

According to the expressing of $\boldsymbol{Y}$ in (6), the corresponding output is changed into $L_1 \times L_2 \times N$ feature graphs as the input for the next time:

$$\boldsymbol{O}_i^l = \left\{\boldsymbol{I}_i^l \cdot \boldsymbol{W}_l^2\right\}_{l=1}^{L_2}. \quad (7)$$

4) *Hash & Bhist*. $\left\{\boldsymbol{O}_i^l\right\}$ acquired in (7) can be treated as the L_2 outputs of PCA filter 1. Performing Hash encoding on these L_2 outputs, we obtain $\left\{\boldsymbol{T}_i^l\right\}$ as:

$$\boldsymbol{T}_i^l = \sum_{l=1}^{L_2} 2^{l-1} H\left(\boldsymbol{I}_i^l \cdot \boldsymbol{W}_l^2\right), \quad (8)$$

where

$$H(\sigma) = \begin{cases} 0, \text{ if } \sigma \leqslant 0, \\ 1, \text{ otherwise,} \end{cases}$$

in (8) denotes the Heaviside step function. The coding length is 2^{L_2}, yielding a total of L_1 coding graphs $\boldsymbol{T}_i^l$. Partitioning it into B blocks, we can determine B histograms defined as $Bhist\left(\boldsymbol{T}_i^b\right)$ $(b \in [1, L_1])$ to describe the frequency of each block at a decimal scale.

5) *Output of the Cascaded Features.* All of the B histograms are cascaded to provide the final general feature $\boldsymbol{F}_i$ as:

$$\begin{aligned} \boldsymbol{F}_i &= ((Bhist(\boldsymbol{T}_i^1), Bhist(\boldsymbol{T}_i^2), \cdots, Bhist(\boldsymbol{T}_i^{L_1}))^{\mathrm{T}} \\ &\in \mathbb{R}^{(2^{L_2})L_1B}. \end{aligned} \tag{9}$$

3.2.2 *Extraction of Locally Enhanced Features*

The extraction of local features is illustrated in Fig.2. We empirically define points P_a, P_b (or symmetrical $P_{a'}a'Pb'$) corresponding to labeled points $1 \sim 4$ in red as well as their symmetrical points $1 \sim 4$ in green in Fig.2(a) due to two different area chosen strategies. One area chosen strategy is to choose the center region according to the situation of points P_a and P_b; another is to choose the rectangle area according to P_a, P_b, and symmetrical $P_{a'}Pb'$. The regions are allocated to the chosen area and then normalized.

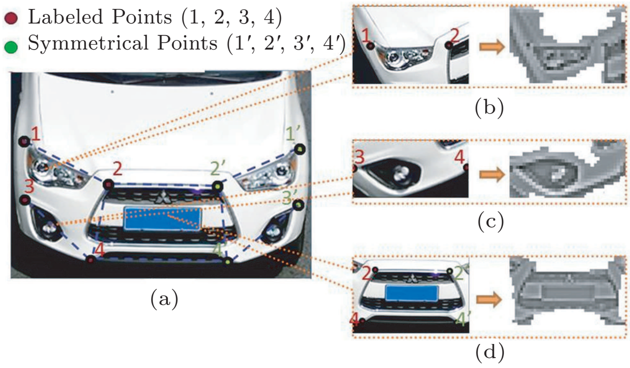

● Labeled Points (1, 2, 3, 4)
● Symmetrical Points (1′, 2′, 3′, 4′)
1
2
3
4
1′
2′
3′
4′
(a)
(b)
(c)
(d)

Fig.2. Extraction of local features in 0°. (a) Important features chosen area in 0°. (b) Local 1: points 1 and 2 are defined as P_a, P_b respectively according to strategy 1. (c) Local 2: points 3 and 4 are defined as P_a, P_b respectively according to strategy 1. (d) Local 3: points 1 and 2 are defined as P_a and P_b respectively, and symmetrical points 2′ and 4′ are defined as $P_{a'}Pb'$ respectively, according to strategy 2.

PCANet-based feature extraction is performed on these regions in the same way at possibly different scales, and the extracted features are denoted with $\{\boldsymbol{f}_{k,i}\}$. We define k as the number of regions that are extracted, and K as the total number of local regions that need to be extracted, and then it is clear that $1 \leqslant k \leqslant K$. Fig.2 shows the features that are the most representative properties of the regions.

3.3 Feature Fusion and Classification

General feature in (9) and local features are fused in a weighted manner. And the weight is for the general feature and β for the local features. Alternatively, the weight for each part can be set to β_k $(1 \leqslant k \leqslant K)$ for the purpose of adapting the weight to the choice of components during the training of models with different angles. Thus, we obtain $\boldsymbol{F}_{\text{full}}$ as:

$$\begin{cases} \boldsymbol{F}_{\text{full}} = \alpha \boldsymbol{F}_i + \beta_k \boldsymbol{f}_{k,i}, \\ \alpha + \sum\limits_{k=1}^{\mathrm{K}} \beta_k = 1. \end{cases} \tag{10}$$

In real-world applications, in order to facilitate the choice of labeling points for the annotator and reduce the amount of labeling work while taking practical factors into account, we abandon some outlying feature points and noise points. Repeated choices may be made for some regions. Details are available in Section 4. Hence, (10) is rewritten into:

$$\begin{cases} \boldsymbol{F}_{\text{full}} = \alpha \boldsymbol{F}_i + \beta_k \boldsymbol{f}_{k,i}, \\ \alpha + \sum\limits_{k=1}^{K} \beta_k \approx 1. \end{cases} \tag{11}$$

Through (11) we can obtain the weighted local features.

Finally, after being cascaded completely, $\boldsymbol{F}_{\text{full}}$ is used as the final feature for classification and input to the softmax classifier.

4 Vehicle Type Recognition Algorithm Based on Locally Enhanced PCANet

Honda ASX and Chevrolet EPICA are chosen here and are denoted with "M" and "S" vehicle types, respectively. Procedures of the proposed algorithm are shown in Fig.3. The steps for feature extraction are elaborated below.

4.1 General Feature Extraction

The image is resized to 200×160 (i.e., $m = 200$, $n = 160$) and partitioned into $k_1 \times k_2$ blocks, and the vertical step and the horizontal step are both set to 10. The block matrix has $(19 \times 19) \times (19 \times 15) = 102\,885$ dimensions. We cut out large blocks to make the general feature more rotation-invariant. The fine-grained

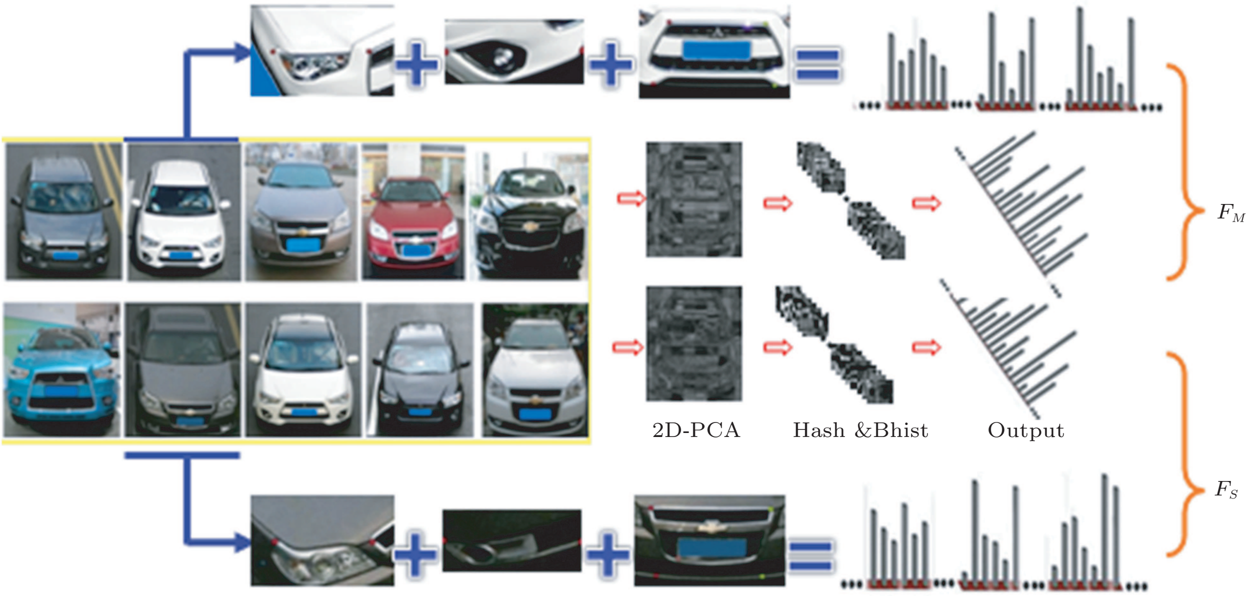


Fig.3. PCANet-based vehicle type recognition. F_M: general feature; F_S: local feature.

features can be effectively complemented through subsequent local features. Note that in the practical operations, in order to restore the image to the original size after it passes through a filter, we need to add zero operation based on the image edge processing and change the image into 210 160 blocks. The block matrix $\boldsymbol{X}$ thus gains $(19 \times 19) \times (20 \times 16) = 115\,520$ dimensions.

We can obtain the first L_1 feature values λ_l $(l = 1, 2, \ldots, L_1)$ of $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$'s orthonormal feature vectors, λ_n as well as their corresponding feature μ_l, based on (3) as the classic method for solving PCA. The covariance matrix $\boldsymbol{S}_{\mathrm{T}}$ is computed as:

$$\boldsymbol{S}_{\mathrm{T}} = \sum_{i=1}^{N} \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}. \tag{12}$$

$\boldsymbol{S}_{\mathrm{T}}$ can be subjected to feature value decomposition based on the principles of SVD[15]. That is, the l-th orthonormal feature λ_n of $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ in (12) can be obtained using the feature value μ_n and the feature vector $\boldsymbol{v}_n$ of $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$.

$$\lambda_i = \frac{1}{\sqrt{\mu_i}} \boldsymbol{X} v_i.$$

The first eight features of λ_l are denoted as $q_l(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})$. According to the following formula:

$$\boldsymbol{W}_{\mathrm{pca}} = \mathrm{argmax}\,|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{T}}\boldsymbol{W}|, \tag{13}$$

we can obtain the feature space matrix $\boldsymbol{W}_{\mathrm{pca}}$ in (13) from $\boldsymbol{I}_i$ to $\boldsymbol{I}_i^l$, i.e., $\boldsymbol{W}_l^1$. Taking (5) into account, we can compute the result of PCA filter 1 as $\{\boldsymbol{I}_i^l\} = \boldsymbol{Y}$, where $\boldsymbol{Y}$ denotes the 8 200 160 features, which are inputted to the PCA filter 2 network.

The method for the second layer of PCA is the same with that for the first layer. To facilitate subsequent binary Hash coding, L_2 is still set to 8, with the other settings remaining unchanged. We extend the 8 200 160 matrices $\{\boldsymbol{I}_i^l\}$ obtained at the first layer into 88 200 160 features matrices $\{\boldsymbol{O}_i^l\}$.

Based on (7), we perform Hash coding on $\{\boldsymbol{O}_i^l\}$, encoding the binary system into a decimal graph. Each decimal encoding graph is converted into 40×40 hist blocks. The repetitious coverage of each block is 50% (overlap = 0.5). The coding graph can be partitioned into 15 hist blocks ($B = 15$), yielding $1\,600 \times 15$ matrices at last. The histogram statistical method is performed on the obtained matrices. In this way, we can obtain B block histograms from 256×15 cascaded blocks.

Finally, all L_1 histograms are cascaded to constitute $256 \times 15 \times L_1 = 30\,720$ dimensional cascading features, yielding the final general feature $\boldsymbol{F}_i$ in (9).

4.2 Local Features Extraction

Reasons for the choice of local features are as follows. First, we expect to improve recognition accuracy by choosing the right local features modeling in different angles. Second, we help to facilitate the user's switch between angle-wise modeling and multi-angle modeling for the purpose of obtaining a list of most probable vehicle types in the applications that enlarge the search scope to reduce misdetection and omission.

We will use the classic FV algorithm[32] to obtain the top 50 most representative feature points. The manually labeled points are displayed with solid red dots, and the automatically symmetrical points are displayed with solid green dots in Fig.4(a) and Fig.4(b), The feature points of the vehicle front images are displayed with hollow red circles in Fig.4(c). The effective regions are chosen under the rule of planning regions that contain more feature points with manually labeled points as few as possible. Few outlying feature points and noise points that are not included will be abandoned. For example, the *i* the vehicle front window shown as region "R_B" in Fig.4(c) has a lot of interferences due to illumination, sticker locations and driver features. Note that the choice of the labeled points and regions differs with the angle during the modeling process, as shown in Fig.4(a).

Choice of the region is illustrated in Fig.2, where the red reference points are labeled in the 600 × 480 0° image and the green labeling points are obtained by mapping along the symmetry axis. Because the feature points of the vehicle's front side are concentrated in the front face around the headlight, we extract four feature points from the left part of this region in Fig.2(a): 1) point 1: left-top-point of the head lamp; 2) point 2: left-top-point of the radiator grille; 3) point 3: left-top-point of the bumper in the middle of the radiator grille; 4) point 4: left-bottom-point of the bottom grille. The labeling points at the right part can be determined symmetrically. To guarantee that the middle region of different vehicle types can be acquired completely, we extract the image of this region using the following method: let P_a and P_b denote the labeling points 1 and 2, respectively; firstly, we choose the middle region and rotate the regional image to the horizontal line at the middle of the two labeling points 1 and 2, and the pixel value between the two points is equal to 20. We leave 5 pixels at the top and bottom sides, and leave 10 pixels at the left and right sides in order to ensure the large-light block is 3 020 in size (i.e., Fig.2(b)). Similarly, we handle the labeling points 3 and 4 in the same way as the labeling points 1 and 2 (i.e., Fig.2(c)), and set the distance between them to 30 pixels. Secondly, we keep 15 pixels at the top and also the bottom, and leave 0 pixel at the left and right to obtain a 3 015 fog-lamp block. Lastly, we focus on points 2 and 4, and symmetrically choose the right part of points 2′ and 4′ in the image as the radiator region of the vehicle, which is then resized to a 6 040 block (i.e., Fig.2(d)).

PCANet-based feature extraction is done on the three regions shown in Fig.2(b), Fig.2(c) and Fig.2(d) respectively. At this time, the region is no longer se-

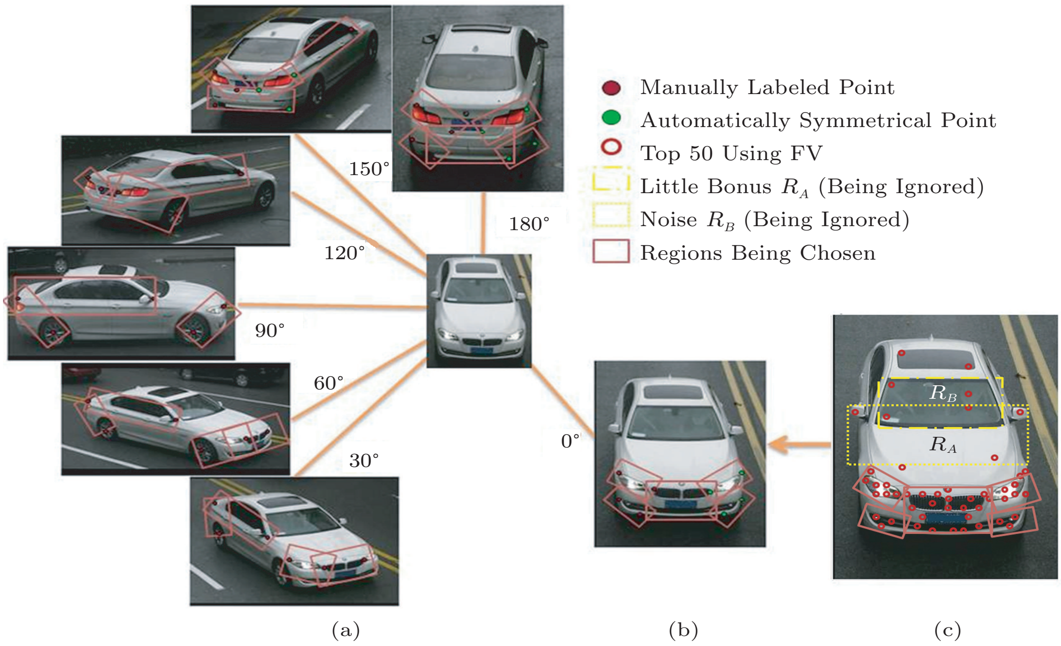


Fig.4. Example of BMW 5 in dataset Sh-vehicles (ShV). (a) Seven angles used to collect the vehicle image and the choice of locally enhanced region at each angle (except 0°). (b) Image of vehicle and the choice of locally enhanced region at 0°. (c) Choices of local features for the 0° image based on the top 50 feature points and the abandonment of related feature points.

lected repetitiously for the last B block, i.e., $overlap = 0$. The steps for obtaining the $6 \times 256 \times 8 + 3 \times 256 \times 8 + 6 \times 256 \times 8 = 30\,720$ local features $\boldsymbol{f}_i$ are skipped here. The local feature values are finally determined and the extracted main features of the local blocks are shown in Fig.2.

4.3 Features Fusion

Based on (11), the ratio of general feature to local features is set to 1:1 by default, i.e., $\alpha = 0.5$. Next, we compute the proportion of the top 50 feature points of each region as the weight of this region to represent the general feature. The proportion of the abandoned top 50 feature points is incorporated into the general weight α again. In the example of vehicle front, we set $\alpha = 0.6$, $\beta_1 = 0.14$, $\beta_2 = 0.06$, $\beta_3 = 0.21$ and then gain weighted $\boldsymbol{F}_{\text{full}}$. Note that these local features rely on different scales and parameters. The combination of the existing general feature and the local features makes the feature values more representative to the scale and dimension of vehicular components. The dimensionality of the local features is almost the same with that of the general feature. Hence, our proposed method is very robust to angles, image qualities and the number of samples. This property will be further discussed in Section 5.

5 Experimental Results and Analysis

5.1 Datasets

In this paper, we choose open datasets Comp[12] and Scar[11], and a large-scale self-built dataset called Sh-vehicles (ShV) to implement relevant experiments and comparisons. The property differences among the three datasets are shown in Table 1, including published year of the dataset (PYear, for short), the time of vehicles made in factory or collected on roads (time, for short), the source of data (source, for short), the scales of the dataset (scale, for short) containing main and minor data compositions (Main-C and Min-C respectively, for short), the number of vehicles types according to the model (MNum), the number of vehicles types according to the year of manufacture (YNum, for short), viewpoints (or angle) division model (angle, for short), and the imbalance problem of samples (imbal, for short).

We firstly evaluate the performance of the proposed algorithm on Comp and Scar in Subsection 5.2 and Subsection 5.3 respectively before we present the experimental results on our own dataset. But because of several differences among Scar, Comp and ShV, the recognition accuracy of the proposed algorithm might be reduced compared with that on ShV. The reason why we design ShV rather than use the existing open datasets is that our project has stricter requirements on vehicle type collection rate and recognition accuracy. Existing datasets based on the Internet are unable to meet these requirements for their small scales, imbalanced samples, too-wide angle divisions, and the lack of standardization of collected data.

ShV consists of images of 1 797 vehicle types according to year of manufacture or 950 vehicle types according to model. Ignoring the variants of the same types designed in different years, the range properties include 0°, 30°, 60°, 90°, 120°, 150° and 180° shown in Figs.4(a) and 4(b). That is, we capture images every 30° as a range from the left 0° to 180° of a slowly driving vehicle, gain 7 images from the vehicle from front, and add the images into the dataset. In our opinion, the 180°∼360° images can be obtained by mapping along the vertical axis. Hence, full-angle vehicle type recognition can be achieved without the need of repeated collection from the right side. We collect 2.5 million images for each vehicle which really ran on roads from 2014 to 2017. Among them, the $200\,000 \times 7$ (1.4 million) full-angle complete images are strictly chosen and

Table 1. Comparison Among Comp[12], Scar[11] and ShV

Dataset	PYear	Time	Source	Scale		YNum	MNum	Angle	Imbal
				Main-C	Min-C				
Comp[12]	2015	Made in 2006∼2014	Web-nature, surveillance-nature	136 727	5 000 (front view)	1 687	431	5 viewpoints for 360°	In viewpoint & types
Scar[11]	2013	Made in 1990∼2000	Internet	16 185 (Car-197)	512 (BMW-10)	None	197+10	5 viewpoints for 360°	In viewpoint & types
ShV	2017	Collected during 2014∼2017 on roads in Shanghai	Surveillances on road	250×10^4	10×10^4	1 797	950	7 angles for 180° or 12 angles for 360°	Only in types

labeled as the training dataset Y_{train}, and the angle offset in a same angle range is less than 5°. Let Y_{test1} denote the remaining one million images with the limitation of incomplete capturing angles and large deviations or partial occlusions. Let Y_{test2} denote the 100 000-plus images that consist of different vehicle sample sizes and noise background extracted from video monitoring images. Y_{test1} and Y_{test2} constitute Y_{test} as the test dataset of our new dataset.

Because we adopt the real-world data, the top 30 most popular vehicle types (occupying 1.7% of all vehicle types, e.g., Volkswagen Santana, Buick Lacrosse) account for 11.7% of all images. As for some large vehicle types that occupy 16% of all vehicle types, their images account for 45.1%. And the images of the first 33% vehicle types account for 73.6%. The changing of partition of samples of more popular types driving on roads (figured as Y axis) with the partition of more popular types (figured as X-axis) is shown in Fig.5. Although the images in ShV are not uniformly classified in types, the samples are perfectly balanced in angles in Y_{train}, which will largely reduce the possibility of the lack of samples in some angles in open datasets, especially Scar.

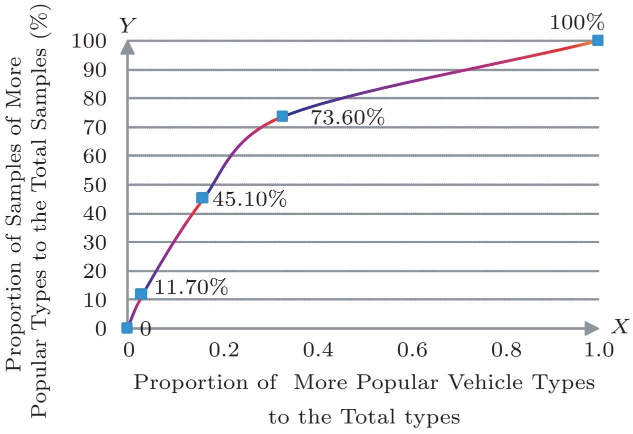


Fig.5. Non-uniform distribution of samples in our database of vehicle types.

5.2 Experiment on Comp[12]

To compare the performances among the overfeat method which is proposed in [12] (overfeat, for short), the PCANet algorithm[20] only using general feature $\boldsymbol{F}_i$ (general PCANet, for short) and our proposed method, we firstly test them on Comp, for Comp has a larger scale and its vehicle models are more similar to vehicles in domestic cities of China. The recognition accuracy comparison among the three methods is shown in Table 2. Due to that the proposed method is separately modeling in different angles, the "angle-wise" value in Table 2 of the proposed method cannot be calculated and thus the average recognition accuracy of angle-wise value is used as a substitute. Numerical results of the experiment denoted with "*(avg)" in figures and tables in this paper are all due to this reason. And they will not be explained separately below.

The results show that the proposed algorithm is slightly inferior to the overfeat method in [12] in terms of the average recognition accuracy, but its recognition accuracy for each angle is much higher than that of general PCANet as shown in Table 2. Table 2 shows that the proposed algorithm achieves the most performance gains at viewpoints F (front), FS (front-side) and R (rear), and its recognition accuracy for F and FS surpasses the angle-wise result of overfeat. The proposed algorithm has greater advantages over the overfeat on the test dataset of distinguishable viewpoints. This further demonstrates the correctness of locally enhanced modeling and also proves the effective label plays an important role in fixing location on vehicles.

In addition, we use the proposed method on Comp to do the misclassification analysis, and then we find an interesting phenomenon that a vehicle make[11-12]("make", means manufacturer) always uses a similar face or body (especially face) it likes to almost every product the vehicle make made. The proportion of errors caused due to the use of same makes to the total number of errors under different viewpoints is

Table 2. Accuracy Result Using Overfeat[12], General PCANet and Our Proposed Method on Comp Database

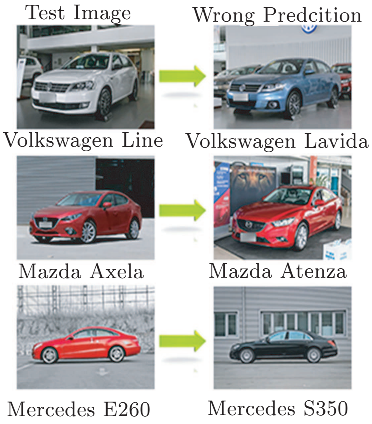
Method	Rank	F	R	S	FS	RS	Angle-Wise
Overfeat[12]	Top-1	0.524	0.432	0.428	0.563	0.598	0.767
	Top-5	0.748	0.647	0.602	0.769	0.777	0.917
General PCANet	Top-1	0.530	0.425	0.404	0.523	0.568	0.714
	Top-5	0.731	0.640	0.593	0.734	0.752	0.847
Proposed	Top-1	0.803	0.709	0.636	0.792	0.757	0.739*(avg)
	Top-5	0.901	0.815	0.761	0.898	0.886	0.852*(avg)

Note: As for the viewpoints, F: front, R: rear, S: side, FS: front-side, RS: rear-side. Using the proposed method: the local F, R and S angles are set to 0°, 90°, 180°, respectively; the FS and the RS angles are set to 30° and 150°, respectively.

shown in Fig.6(a). And wrong samples are also shown in Fig.6(b). This is consistent with the conclusion that "most of the wrong predictions belong to the same car make as the test images" in [12].

Viewpoint	Wrong Proportion Due to the Same Make
F	0.833
FS	0.715
S	0.602
R	0.657
RS	0.674
*(avg)	0.6962

(a)



(b)

Fig.6. Misclassification analysis on Comp using proposed method. (a) Proportion for misclassification due to the same make (manufacturer) using the proposed method in different viewpoints. (b) Examples of misclassification due to the same make.

5.3 Experiment on Scar[11]

The main component of Scar is car-197 for general fine-grained classes. Compared with several algorithms with higher recognition rates in [13], the performance of our algorithm is in line with our expectations as shown in Table 3.

Table 3. Comparison of Recognition Rates Training on Car-197 and BMW-10 of Scar[11]

Method	Accuracy in Car-197 (%)	Accuracy in BMW-10 (%)
BB[13]	92.60	58.70
BB-3D-G[13]	94.50	66.10
SPM[13]	84.50	58.30
SPM-3D-L[13]	85.70	58.70
General PCANet	87.30	57.90
Proposed	92.10* (avg)	71.60* (avg)

Note that the Scar dataset[11] has another small-sample subset of BMW-10, which contains 512 images to evaluate the performance of recognizing non-distinguishable objects. The 3D method produces the highest accuracy of 66.1%[13]. The proposed algorithm is applied to this dataset and achieves an accuracy of 71.6%. Detailed comparison results on BMW-10 are also shown in Table 3. This result indicates that introducing local features to the proposed algorithm is very effective in recognizing high-similarity objects. Fig.7 shows the samples of top 5 wrong predictions and their error rates using the proposed algorithm.

Top n	Pair Name	Test Image	Wrong Prediction	Wrong Proportion
1	BMW5 BMW3			16.9%
2	BMW5 BMW7			9.4%
3	M5 M3			7.2%
4	BMW5 BMW6			6.7%
5	BMW6 BMW7			5.9%

Fig.7. Top 5 wrong predictions of vehicle pairs using the proposed method on Scar.

5.4 Experiment on ShV

5.4.1 *Comparison with Other Algorithms*

We apply various kinds of pattern recognition and neural network training algorithms on Y_{train}. The average recognition accuracy of the proposed algorithm for seven different angle ranges is highlighted in Table 4. From this table, it can be seen that our recognition accuracy is higher than that of the overfeat algorithm of [12] and slightly smaller than the GoogleNet v1 algorithm of [33]. But the operations number of the method in [33] is equivalent to 5.35 GB floating-point operations for one picture but merely 0.06 GB for one picture using PCANet. Hence, its time consumption and computational complexity is much less than that of the CNN like GoogleNet V1.

Table 4. Comparison of Recognition Rates on Y_{train} of ShV

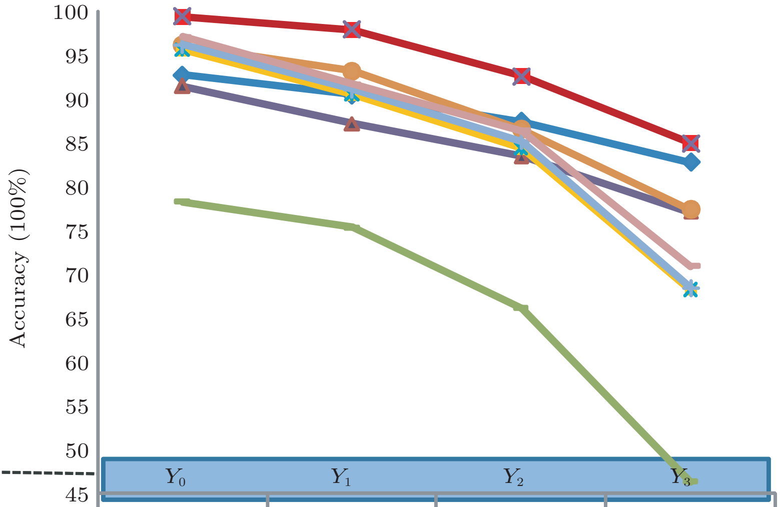
Method	Accuracy (%)
HOG[26]	72.68
FV[32]	73.53
General PCANet	89.20
Proposed	93.28* (avg)
Classic 5-layer handwriting	68.93
Krizhevsky CNN[18]	89.94
Overfeat[12]	91.31
GoogleNet V1[33]	93.62

In order to study the influence of the number of samples on the experimental results, we perform an experiment using the data merge operations used in [12]. That is, we adopt the first 431 popular types and then mix them with the variants of the same sorts to generate the training subset. We compare the recognition rates achieved with different subsets of 0° that consist of different numbers of samples (test dataset Y_{test} also makes corresponding changes according to the training dataset). The results are shown in Fig.8 where N is the number of vehicle types chosen, and i is the number of images for each vehicle type. The traditional pattern recognition method is very effective when the number of samples is small, but its performance deteriorates in the case of many samples. CNN is suitable for large-scale sets of samples, and its performance may improve with the increase of the number of samples. But the fine-grained intra-class classification process involves the design of more PCA filters for the neural network. This will inevitably cause exponentially increased computational complexity and prolonged training time. Compared with traditional pattern recognition and neural network schemes, our proposed method is always robust to any size of sample database and its performance is particularly excellent for the multi-class medium-sized sample database.

5.4.2 *Full-Angle Modeling vs Angle-Wise Modeling: Role of Local Features*

According to the experimental results in [12], the CNN algorithm recognizes the vehicle types more accurately than the angle-wise models after modeling the full-angle images. Is this also true for PCANet? Table 5 compares the overfeat algorithm[12], general PCANet, and the proposed algorithm on ShV. Unlike the viewpoints division model used in [12], we replace the original 5 viewpoints with seven different angle ranges. The results show that the recognition accuracies in the angle of 0°, 30°, and 180° are higher than those of the other four left angles, and that the recognition accuracies in the angles near 90° are slightly higher than or comparable to the full-angle recognition accuracy without considering the local features. This means that the front head and back of the vehicle have the most fine-grained features. The recognition accuracy of the vehicle side is largely dependent on the general feature like the contours. This result enables us to choose the angle-wise modeling method that provides greater recognition

Data	$N \times i$
Y_0	30 × 700
Y_1	150 × 450
Y_2	310 × 150
Y_3	950 × 50



	Y_0	Y_1	Y_2	Y_3
HOG[26]	92.72%	90.41%	87.36%	82.73%
FV[32]	91.38%	87.17%	83.39%	77.06%
General PCANet	96.04%	93.10%	86.47%	77.34%
Proposed	99.37%	97.89%	92.56%	84.93%
Classic 5-layer handwritting	78.23%	75.28%	66.13%	46.31%
Krizhevsky CNN[18]	95.52%	90.43%	84.29%	68.20%
Overfeat[12]	96.12%	91.02%	85.11%	68.45%
Googlenet V1[33]	97.03%	91.62%	86.32%	70.89%

Fig.8. Variation of recognition rates with the sample 0° on different subsets of ShV using different algorithms.

Table 5. Comparison of Experimental Results for Different Algorithms on Y_{train} of ShV

Method	Rank	0	30	60	90	120	150	180	Angle-Wise
Overfeat[12]	Top 1	0.949	0.945	0.936	0.896	0.909	0.917	0.923	0.926
	Top 5	0.960	0.967	0.969	0.937	0.942	0.939	0.942	0.956
General PCANet	Top 1	0.914	0.912	0.900	0.874	0.883	0.904	0.908	0.892
	Top 5	0.947	0.945	0.942	0.932	0.920	0.934	0.936	0.918
Proposed	Top 1	0.971	0.954	0.933	0.892	0.905	0.916	0.927	0.933*(avg)
	Top 5	0.991	0.972	0.955	0.922	0.930	0.942	0.943	0.951*(avg)

accuracy while we could accurately label the angles of the test dataset.

5.4.3 *Correctness of the Local Features Selection*

By assigning various N and i shown in legend at the top left of Fig.8, we build Y_0, Y_1, Y_2, Y_3 as subsets from ShV respectively. It can be seen from the analysis above that compared with the algorithms that only use the PCANet scheme and adopt the same parameter settings, the proposed algorithm based on enhanced general and local features is more accurate by 3%∼8% in different subsets of Y_0, Y_1, Y_2, Y_3. Do we choose the correct local features? To verify this, we test the variation of the recognition accuracy with different combinations of local features on dataset Y_1. The results are shown in Table 6. It can be observed that the front lamp, the fog lamp and the front grid of vehicles shown in the table represent the main features of the front face of vehicle types. The region R_A does not make many contributions to feature extraction; the region R_B renders the general feature irrelevant and reduces classification accuracy. R_A and R_B are showed in Fig.4(c). The local features include the main feature points of most components. The scale of these local features is different from that of the general feature. And the different scales well complement the defects of the global algorithm in terms of classifying fine-grained features, thereby achieving a high recognition rate of 97.89%. As shown in Fig.8, in the case of small-sample multi-class scenario of Y_2 and Y_3, small-scale local features become an important factor in inhibiting decrease in recognition rate, and they make more contributions than the large-scale general feature.

Table 6. Influence of the Choice of Local Components on the Recognition Rate on Y_1 of ShV

Method	Accuracy (%)
Proposed	97.89
General PCANet	93.10
J	84.56
+Front lamp	94.70
+Fog lamp	94.14
+Front grid	95.31
+Engine cover	93.36
+Windscreen	89.73

Note: J: only the local features of the previous three components; +front lamp: general PCANet used the local feature of the vehicle's front lamp; +front grid: general PCANet used the local feature of the vehicle's front grid; +engine cover: general PCANet used the local feature of the vehicle's engine front cover; +Windscreen: general PCANet + fused the local feature of the vehicle's front windscreen.

5.4.4 *Test on Correctness of Angle Classification*

Is our angle classification scheme appropriate? To answer this question, our first experiment is to check whether the recognition accuracy is satisfactory when the angle deviation of the test samples exceeds 30° compared with the modeling samples at the angle of 0°. Next, we study whether modeling on samples of other angles can gain similar performance, for example modeling on 60° samples. Finally, we determine the optimal angle classification scheme.

We firstly choose 50 sets of vehicle images (50×7 images) that are captured at an angle of 0°∼30° and then rotate them at a step length of 5° to constitute a training set Y_1. Variation in the recognition rate is shown in Fig.9(a), where the former is the general PCANet algorithm that does not incorporate locally enhanced features, and the latter is the proposed algorithm. It can be found that the general PCANet is very robust to variation in angles, as the recognition rate remains almost the same when the angle ranges from 0° to 5°. But the recognition rate deteriorates quickly after the angle exceeds 15°. The small-scale local features make no contribution to the final results when the angle is 30°. This means that the large-scale general feature contributes enormously to the robustness of the proposed algorithm against variation in angles. Hence, it is verified that the large-scale block largely contributes to the feature's rotation invariance.

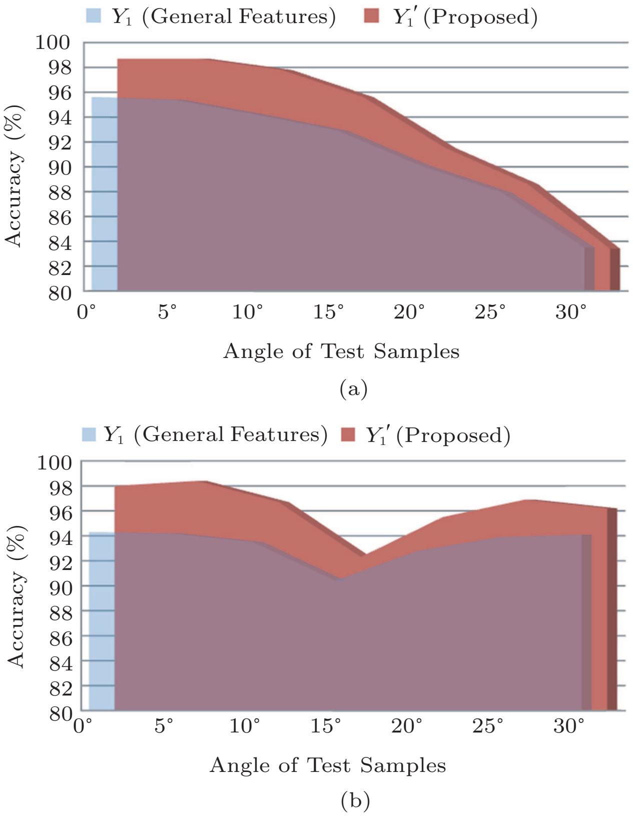


Fig.9. Variation of recognition rates on Y_1 and Y_1' of ShV as a function of angle deviation.

The 45° classification scheme in [15] is more unable to recognize satisfactorily. To check the Y_1 set in angle of 0° is not accidental, thus we model samples at 60°, and incorporate the 15°∼105° vehicle images into the test dataset. The results are shown in Table 7. This table shows that our angle range classification scheme is appropriate.

Table 7. Comparison of Recognition Rates with Actually Measured Angle Deviations Modeling on 60° Samples of ShV

Angle	Accuracy (%)
15°	56.69
30°	79.17
45°	89.16
60°	93.34
75°	84.97
90°	71.21
105°	53.21

Considering the fact that the recognition rate decreases quickly after the angle exceeds 15°, we enrich the original dataset Y_1 with the images of the same vehicle types captured at an angle of 30°. That is, the original dataset Y_1 is enlarged to a set of Y_1', $150 \times 450 \times 2$ in size. The variation of recognition rate as a function of angle for Y_1' is shown in Fig.9(b). It is discovered that after the addition of the 30° images, the highest level of recognition rate achieved at an angle 0°∼10° decreases by 1.3%. And the recognition rate at an angle of 13°∼17° exhibits a trough around 3%. But in the case of angle equal to 20° and 30°, the recognition rate increases from 88.6% and 83.4% to 96.9% and 96.2%, respectively. These results indicate that adding multi-angle samples to the regionally clear images that are captured at an angle of 30° at the face side can generally make the enlarged-modeling method (the method that adds adjacent angle samples to Y_1 dataset to build enlarged Y_1') less sensitive to angle deviations. Thus, the enlarged-modeling method is worthy of our choice. We extend the enlarged-modeling method by adding the 60° samples (local regions cannot be extracted for use, and it only refers to general PCANet). As a result, the angle interval gaining high recognition rate is reduced and the overall recognition rate is reduced to less than 50%. This indicates that the enlarged-modeling method can only be used in the case of small angle deviation, and cannot obtain satisfactory recognition rate in each angle or larger angle deviation. This experiment further proves the correctness of our angle classification scheme.

6 Conclusions

In this paper, we proposed a multi-scale PCANet method based on the enhancement of local features, and also developed a large and standard multi-class dataset ShV. After comparing the classification performance of the proposed method with that of other methods on datasets with varying orders of magnitude, i.e., Comp[12], Scar[11], and ShV, we demonstrated the correctness of angle-wise modeling, choice of local features, and angle classification. The experimental results on Scar, Comp and ShV showed that the proposed algorithm is superior to the other methods. Although the proposed algorithm needs to label the fixed point of test samples in an angle-wise manner, the amount of labeling work is very small and the labeling points are very distinguishable. Compared with the traditional pattern recognition algorithms which feature a large number of labeling points and gain low recognition accuracy and with the multi-level deep learning algorithms that need many samples and consume heavy computational resources, the proposed algorithm is the chief choice for applications where objects from large and standardly classified datasets like ShV can be recognized accurately and efficiently. Note that if the applications place small demand on recognition accuracy, the angles are not distinguishable in the dataset and the annotator is not available, then the full-angle model with the recognition scheme without considering local features is a better option. This is because the error of manual labeling and the amount of labeling work can be reduced. The CNN algorithm is recommended if a large number of samples are available and a long period of training time is affordable.

Given more samples, we plan to improve the convenience of the proposed algorithm in the future and add a small number of network layers to implement full-angle labeling-free and adaptive recognition.

References

[1] Simonyan K, Parkhi O, Vedaldi A *et al.* Fisher vector faces in the wild. In *Proc. Conf. British Machine Vision*, September 2013.

[2] Berg T, Belhumeur P N. POOF: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2013, pp.955-962.

[3] Cao Q, Ying Y, Li P. Similarity metric learning for face recognition. In *Proc. IEEE Int. Conf. Computer Vision*, January 2013, pp.2408-2415.

[4] Sun Y, Wang X, Tang X. Deep learning face representation from predicting 10 000 classes. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2014, pp.1891-1898.

[5] Sun Y, Chen Y, Wang X *et al.* Deep learning face representation by joint identification-verification. In *Proc. Int. Conf. Neural Information Processing Systems*, November 2015, pp.1988-1996.

[6] Feris R S, Siddiquie B, Petterson J *et al.* Large-scale vehicle detection, indexing, and search in urban surveillance videos. *IEEE Trans. Multimedia*, 2012, 14(1): 28-42.

[7] Hu C, Bai X, Qi L *et al.* Learning discriminative pattern for real-time car brand recognition. *IEEE Trans. Intelligent Transportation Systems*, 2015, 16(6):3170-3181.

[8] Grauman K, Crandall D, Parikh D *et al.* Discovering localized attributes for fine-grained recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2012, pp.3474-3481.

[9] Wah C, Horn G V, Branson S *et al.* Similarity comparisons for interactive fine-grained categorization. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2014, pp.859-866.

[10] Goering C, Rodner E, Freytag A *et al.* Nonparametric part transfer for fine-grained recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2014, pp.2489-2496.

[11] Krause J, Deng J, Stark M *et al.* Collecting a large-scale dataset of fine-grained cars. In *Proc. the 2nd Fine-Grained Visual Categorization Workshop*, June 2013.

[12] Yang L, Luo P, Chen C L *et al.* A large-scale car dataset for fine-grained categorization and verification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2015, pp.3973-3981.

[13] Krause J, Stark M, Deng J *et al.* 3D object representations for fine-grained categorization. In *Proc. IEEE Int. Conf. Computer Vision*, June 2013, pp.554-561.

[14] Lin Y L, Morariu V I, Hsu W *et al.* Jointly optimizing 3D model fitting and fine-grained classification. In *Proc. European Conference on Computer Vision*, September 2014, pp.466-480.

[15] Stark M, Krause J, Pepik B *et al.* Fine-grained categorization for 3D scene understanding. In *Proc. Conf. British Machine Vision*, September 2012, pp.228-236.

[16] Sochor J, Herout A, Havel J. BoxCars: 3D boxes as CNN input for improved fine-grained vehicle recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2016, pp.3006-3015.

[17] Zhang X, Zhou F, Lin Y *et al.* Embedding label structures for fine-grained feature representation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2016, pp.1114-1123.

[18] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In *Proc. Int. Conf. Neural Information Processing Systems*, November 2012, pp.1097-1105.

[19] He H, Shao Z, Tan J. Recognition of car makes and models from a single traffic-camera image. *IEEE Trans. Intelligent Transportation Systems*, 2015, 16(6): 3182-3192.

[20] Chan T H, Jia K, Gao S *et al.* PCANet: A simple deep learning baseline for image classification? *IEEE Trans. Image Processing*, 2014, 24(12): 5017-5032.

[21] Dong Z, Wu Y, Pei M *et al.* Vehicle type classification using a semi supervised convolutional neural network. *IEEE Trans. Intelligent Transportation Systems*, 2015, 16(4): 2247-2256.

[22] Xie S, Yang T, Wang X *et al.* Hyper-class augmented and regularized deep learning for fine-grained image classification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2015, pp.2645-2654.

[23] Zhao B, Wu X, Feng J *et al.* Diversified visual attention networks for fine-grained object classification. *IEEE Trans. Multimedia*, 2017, 19(6): 1245-1256.

[24] Zia M Z, Stark M, Schindler K. Towards scene understanding with detailed 3D object representations. *International Journal of Computer Vision*, 2015, 112(2): 188-203.

[25] Arandjelovic R, Zisserman A. Three things everyone should know to improve object retrieval. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2012, pp.2911-2918.

[26] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2005, pp.886-893.

[27] Chen T, Chen Z, Shi Q *et al.* Road marking detection and classification using machine learning algorithms. In *Proc. Intelligent Vehicles Symp.*, June 2015, pp.617-621.

[28] Wang X S, Cai C. Weed seeds classification based on PCANet deep learning baseline. In *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, December 2015, pp.408-415.

[29] Wu J, Shi J, Li Y *et al.* Histopathological image classification using random binary hashing based PCANet and bilinear classifier. In *Proc. Conf. European Signal Processing*, August 2016, pp.2050-2054.

[30] Xia Y, Li J, Qi L *et al.* Loop closure detection for visual SLAM using PCANet features. In *Proc. Int. Conf. Neural Networks*, July 2016, pp.2274-2281.

[31] Jia H, Sun Q, Wang T. PCANet for blind image quality assessment. In *Proc. Int. Conf. Computational Intelligence and Security*, December 2015, pp.195-198.

[32] Kwang K, Keechul J, Hang J K. Face recognition using kernel principal component analysis. *IEEE Signal Processing Letters*, 2002, 9(2): 40-42.

[33] Szegedy C, Liu W, Jia Y *et al.* Going deeper with convolutions. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2014.

Qian Wang is a senior engineer in Information Center of Criminal Investigation Department of Shanghai Public Security Bureau, Shanghai, and also a Ph.D. candidate of School of Computer Engineering and Science, Shanghai University, Shanghai. Her research interests mainly include intelligent surveillance analysis, deep learning neural network, and intelligent human biological characteristic identification.

You-Dong Ding received his Ph.D. degree in computational mathematics from University of Science and Technology of China, Hefei, in 1997. He is now a professor, doctoral supervisor, and the vice dean of Shanghai Film Academy, Shanghai University, Shanghai. He is also the vice director of Shanghai Engineering Research Center of Motion Picture Special Effects, Shanghai. His main research areas include computer graphics, digital visual media technology and film big-data processing, etc. He is a senior member of CCF.