Vehicle Logo Recognition System Based on Convolutional Neural Networks With a Pretraining Strategy

Yue Huang, Ruiwen Wu, Ye Sun, Wei Wang, and Xinghao Ding, *Member, IEEE*

***Abstract*—Since a vehicle logo is the clearest indicator of a vehicle manufacturer, most vehicle manufacturer recognition (VMR) methods are based on vehicle logo recognition. Logo recognition can be still a challenge due to difficulties in precisely segmenting the vehicle logo in an image and the requirement for robustness against various imaging situations simultaneously. In this paper, a convolutional neural network (CNN) system has been proposed for VMR that removes the requirement for precise logo detection and segmentation. In addition, an efficient pretraining strategy has been introduced to reduce the high computational cost of kernel training in CNN-based systems to enable improved real-world applications. A data set containing 11 500 logo images belonging to 10 manufacturers, with 10 000 for training and 1500 for testing, is generated and employed to assess the suitability of the proposed system. An average accuracy of 99.07% is obtained, demonstrating the high classification potential and robustness against various poor imaging situations.**

***Index Terms*—Convolutional neural networks (CNNs), deep learning, pretraining, vehicle logo recognition (VLR).**

I. Introduction

VEHICLE license plate location (LPL) [1], [2] and vehicle manufacturer recognition (VMR) [3] play a crucial role in intelligent transportation systems [4]. These systems are useful in providing car ownership statistics to governments and businesses. In addition, manufacturer recognition can assist in vehicle identification, e.g., vehicle license plates are often replaced in stolen cars. Since a vehicle logo is the clearest indicator of a vehicle manufacturer, VMR systems always employ logo information for recognition.

Manuscript received May 7, 2014; revised September 28, 2014 and December 3, 2014; accepted December 22, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 30900328, Grant 61172179, Grant 61103121, Grant 71103150, and Grant 81301278; by the National Key Technology Research and Development Program under Grant 2012BAI07B06; by the Fundamental Research Funds for the Central Universities under Grant 2013121023; and by the Research Fund for the Doctoral Program of Higher Education under Grant 20120121120043. The Associate Editor for this paper was P. Cerri. *(Corresponding author: Xinghao Ding.)*

Y. Huang, R. Wu, Y. Sun, and X. Ding are with the Department of Communication Engineering, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China (e-mail: dxh@xmu.edu.cn).

W. Wang is with the Department of Electronic Engineering, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2014.2387069

Compared with LPL, which has been extensively studied and addressed in research and industry, there have been fewer studies contributing to vehicle logo recognition (VLR). VLR is still a challenge in pattern recognition and computer vision due to its requirements for precise logo segmentation, robustness against various imaging situations, and real-time implementation. According to the literature in this area, existing logo-based VMR systems always consist of two stages, i.e., precise vehicle logo detection (VLD) and VLR. Wang *et al.* [5] detected vehicle logos using edge features, and then, they implemented logo recognition using template matching and edge orientation histograms. Although they achieved an accuracy of 90%, their approach is still limited in complicated environments such as those with shadows and light reflections. Sam and Tian [6] presented a solution for VLR using a method called "Modest AdaBoost" combined with radial Chebyshev moments to address viewpoint variation, achieving an accuracy of 92%. However, the computation time is more than 2 s for a single image, which is not suitable for real-time applications. Dlagnekov and Belongie [7] utilized scale-invariant feature transform (SIFT) features to identify the vehicle manufacturer and model on rear-view vehicle images, but the system performance does not meet real-time requirements. Psyllos *et al.* [8] also proposed a SIFT-based enhanced matching scheme, which detects a logo through a phase congruency feature map method. This scheme seems promising but suffers from illumination variations or a wide range of viewpoints. Yu *et al.* [9] proposed a system for VLR based on the "Bag-of-Words" model, which uses a dense SIFT to extract stable features, quantize features by soft assignment, and compute a histogram with spatial information to improve performance. Vehicle logo images are represented as histograms of visual words and then classified by a support vector machine (SVM). The reported recognition accuracy is 97.3%, and the computing time is 22 ms for each image in recognition. Dai *et al.* [10] adopted Chebyshev moment invariants to extract the six eigenvectors of a vehicle logo, and an SVM is applied to the binary image recognition. The algorithm is reported to be robust to noise; the recognition accuracy is 92%; and the computing time is 1 s for each image.

However, previous logo recognition in VMR is still limited by: 1) the use of hand-crafted features (e.g., a SIFT), which is inadequate to simultaneously address various imaging situations such as poor illumination, logo rotation, viewpoints variation, and noise; and 2) strong dependence on precise logo detection since inaccurate logo detection will lead to a dramatically decreased recognition rate.

Convolutional neural networks (CNNs) are hierarchical neural networks whose convolutional layers alternate with subsampling layers, which are reminiscent of simple and complex cells in the primary visual cortex of the brain [11]–[13]. Motivated by the success of CNNs in machine vision problems, this paper proposes a novel VLR system based on a CNN. Rather than precise logo detection and segmentation, this method coarsely segments a large region of an image, which can be very efficiently detected using a simple assumption based on the location information of LPL. Higher order features can be then directly extracted based on the stacked trainable stages in the CNN using repeating convolutions, nonlinear mapping, and max pooling the large segmented region from the previous step. Finally, logo recognition is implemented as the final layer in the CNN with a supervised backpropagation (BP) neural network classifier [14].

Although a CNN is a good option for VLR, this method still has some limitations. In the convolutional layer at each trainable stage, the kernels/weights employed in the convolution are trained by BP neural networks, which is very time consuming. For example, in this proposed system, it took approximately 15 h to train all kernels from the 10 000 training images. This has limitations in the real-world applications of logo image recognition, where testing samples will become training samples after recognition; thus, the training data will be updated frequently. Therefore, a CNN-based logo recognition system will be very limited since the kernels in the neural network need to be trained with the frequently updated training samples. In addition, network training for classification is critically dependent on the expertise of parameter tuning and some *ad hoc* tricks.

To address this problem of CNN-based logo recognition, a principal component analysis (PCA)-based pretraining strategy has been introduced. This strategy can improve the performance of previous CNN-based systems by efficiently and effectively estimating the kernels in the convolutional layer, without tedious tuning by the BP neural network, thus dramatically reducing the computational cost of kernel training. The improved system is a better option for real-world applications since the system is able to enhance the classification accuracy and reduce the training time simultaneously.

The contributions of the proposed system can be summarized as follows. First, a PCA-based pretraining strategy has been suggested for a CNN logo recognition model to dramatically reduce the computational cost of training procedures, thus ensuring that the proposed models satisfy the requirement of logo recognition for efficient training procedures since the training samples are updated frequently. Second, rather than hand-crafted features, the proposed system automatically extracts features that are robust enough to provide satisfactory classification accuracy with various outdoor imaging situations. Finally, a coarse segmentation approach is proposed based on a very simple assumption; thus, precise logo detection is no longer required.

This paper is organized as follows. VLR systems based on a CNN and a pretraining strategy are described in Section II. In Section III, the generation of the logo data set is presented, and experimental comparisons and robustness validations are given. Finally, Section IV concludes this paper.

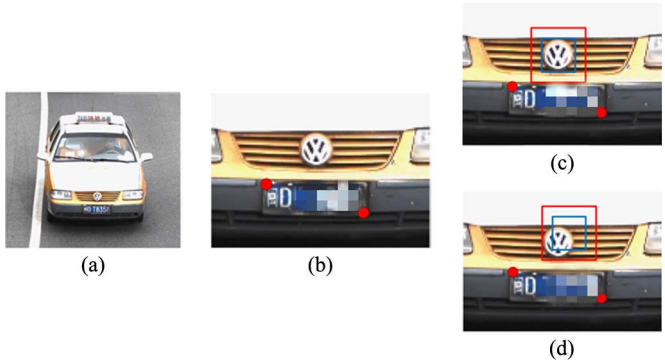


Fig. 1. Proposed car logo detection: the images are captured from the monitoring system in (a), and then, the license plate is identified for each image in (b). Instead of the precise detection in previous studies [the blue box in (c) and (d)], the proposed coarse segmentation detects a much larger area [the red box in (c) and (d)] in order to increase the accuracy of logo detection for the subsequent recognition.

II. Framework of Logo-Based VMR System

A. Coarse Segmentation

Both the proposed VLD scheme and a traditional logo detection system are presented in Fig. 1 to assist with the description of the technique used. The images are first captured from the monitoring system [see Fig. 1(a)], and the vehicle license plate is then identified for each image using the embedded LPL module in the monitoring system. The coordinates of the upper left and lower right corners of the vehicle plate are identified after LPL, as shown in the closeup image in Fig. 1(b). The traditional logo recognition system then carefully detects the vehicle logo and removes the background around the logo, including the radiator and the grille [the blue box in Fig. 1(c) and (d)]. The proposed VLD scheme coarsely segments a large region containing the vehicle logo above the vehicle license without removing backgrounds [the red box in Fig. 1(c) and (d)].

The segmented region [the red box in Fig. 1(c) and (d)] is much larger than that in the traditional logo recognition system [the blue box in Fig. 1(c) and (d)] since it is just an approximate estimation of the vehicle logo's location. In this paper, we consider VLR applications specifically for mainland China, which is currently experiencing a wave of urban expansion. According to the Road Traffic Safety Policy in mainland China, every vehicle logo must be located above the vehicle license plate, and the vehicle license plate must be the same size across all cars.

This coarse segmentation for the proposed logo recognition can be very efficiently performed by making the very simple assumption that the vehicle logo is always located within a certain area above the license plate. The vehicle logo can be then assumed to be located within any part of the segmented area, removing the requirement for image matching or the mask design used in previous studies.

If the VLD scheme is accurate, as shown in Fig. 1(c), both the proposed method and the traditional logo recognition method can achieve similar recognition rates. However, if the VLD scheme is inaccurate, e.g., part of the logo is outside the detected region such as the example shown in Fig. 1(d), this will lead to a serious decline in the recognition rate since traditional VMR systems are highly dependent on the detection

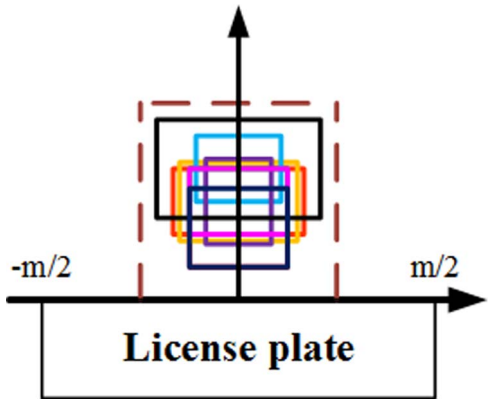


Fig. 2. Coarse segmentation (dashed brown box with a size of $m/2 \times m/2$) is defined based on the fact that all logos in this paper are located within this area (VW: red; Honda: orange; Toyota: yellow; Cherry: yellow; Peugeot: blue; Buick: purple; Hyundai: rose; Citeron: black; Lexus: dark blue).

Fig. 3. Examples of coarse segmentation with (a) correct LPL and (b) incorrect LPL.

and segmentation procedure. By contrast, the proposed system only crudely extracts a larger region for recognition from the LPL data, thus ensuring a very high probability that the logo will be located inside the detected region. Preprocessing to obtain precise VLD is no longer required in the presented logo recognition technique.

As shown in Fig. 1(c) and (d), coarse segmentation is defined as a particular region above the license plate, which can be easily implemented. After LPL, given that the size of the license plate is $m \times n$, $(m > n)$, then the segmented image is assumed to be located at the top center of the license plate, with size $b \times b$, where $b = 1/2m$. It should be emphasized that we are only considering the top ten popular cars in mainland China; suburban utility vehicles (SUVs) and other larger vehicles are not considered in this paper. We analyzed vehicle logo locations relative to the detected license plate for the ten vehicle manufacturer logos. As shown in Fig. 2, the proposed coarse segmentation (dashed brown box) will encapsulate all the vehicle logos (boxes in different colors) considered in the proposed system. In addition, although the LPL is incorrect, the logos still have a large probability of being within the coarse segmented area, as shown in Fig. 3. The detected regions can be also defined with a larger size to accommodate different kinds of vehicles.

B. VLR With CNN

VLR via a classical CNN is presented in this section. It has been observed that logos are objects in the images, which can be decomposed into motifs, and the motifs can be further decomposed into edges [15]. A CNN, consisting of multiple trainable stages stacked on top of each other, is employed to extract the features hierarchically, as in Fig. 4 [16]. The input is the detected images that are 70×70 pixels and contain the vehicle logo, which is obtained from coarse segmentation, as described in the previous section. We will take layer $C1$ and layer $S2$ as examples to explain the flowchart since the following layer, e.g., $C3$ and $S4$, will repeat the same procedures.

$C1$: Based on convolving the input image with different kernels (or weights), several feature maps can be generated in layer $C1$, which are denoted as $C1_i$, $i = 1 : N$, where N is the number of kernel as

$$C1_i = (\text{sigmoid}(w_i \otimes x)) \tag{1}$$

where x is the original image with size $s_x \times s_y$, $\otimes$ denotes the convolutional operation, w_i corresponds to the ith convolutional kernel, and nonlinear mapping sigmoid(x) is defined as $f(x) = 1/(1 + e^{-x})$. Filter w_i is randomly initialized at first and is then trained with a well-known BP neural network [13], [16].

$S2$: Each feature map in $S2$ is obtained by a pooling operation called max pooling that is performed on the corresponding feature map in layer $C1$ as

$$S2_i = \text{pool}(C1_i). \tag{2}$$

The maximum activation over nonoverlapping rectangular regions of size (k_x, k_y) is implemented for input $C1_i$. In this paper, $k_x = k_y = 2$. A max-pooling action creates position invariance over larger local regions and downsamples the input image by a factor of k_x and k_y along each direction, producing each feature map in layer $S2$ with size $M_x/2$ and $M_y/2$. Pooling layers are introduced to detect the maximum response of the generated feature maps to different kernels w_i and to reduce the resolution of the feature map, which will extract the multiscale features of logo images. The pooling action also provides built-in invariance to small shifts and distortions, which can happen in various image monitoring conditions. The convolution and max-pooling procedures in layer $C3$ and layer $S4$ are the same as in layer $C1$ and layer $S2$, except with a different kernel size. The convolutional layers and the max-pooling layers alternatively repeat to construct a multilayer structure for feature extraction, where the output of the previous layer is the input of the current layer as

$$S_4 = \text{pool}\left(w_i \otimes (\text{pool}(w_i \otimes x))\right). \tag{3}$$

Since the primitive feature detectors that are useful on one part of the image are likely to be useful across the entire image, units in a layer are organized in planes within which all the units share the same set of weights. By sharing the same set of weights, the CNN have a shift-invariant property and can achieve excellent performance in logo recognition tasks with a range of various imaging conditions. Detailed information of feature extraction by multilayer CNNs is available in [13] and [17].

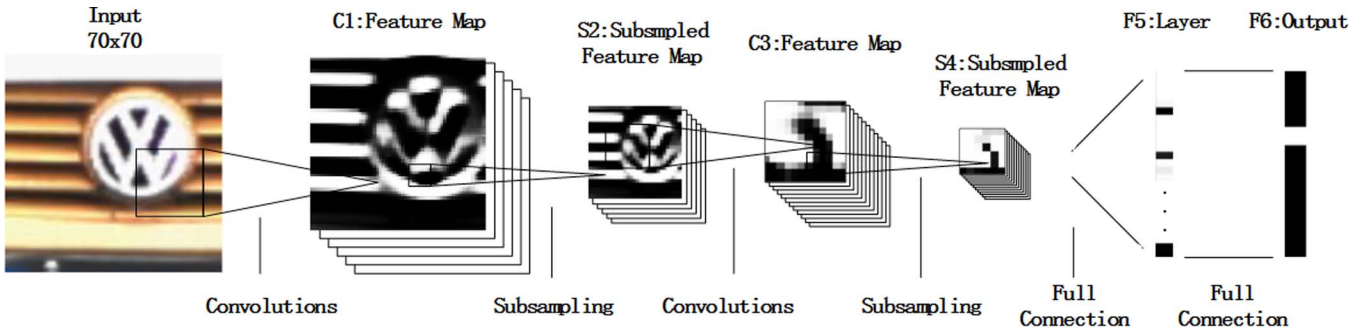


Fig. 4. Architecture of a CNN for VLR.

For the final recognition step, the local features from the input image are combined with the subsequent layers in order to obtain higher order features after the final max-pooling layer $S4$. These higher order features are eventually encoded into a 1-D vector, which is then categorized by a BP neural network classifier in the last layer of the CNN structure.

The CNN feature detector can be supplied with raw pixels to automatically learn low-level and midlevel features of stacked structures, alleviating the need for hand-engineered features and improving the recognition accuracy [17]. In addition, in contrast to traditional fully connected neural networks, the CNN forces the extraction of local features by placing a restriction that the receptive fields of hidden units should be local, which is based on the fact that images have strong 2-D local structures. This restriction thereby reduces the calculation scale and increases the robustness of the system for rotation and translation, which is a technique that is particularly suitable for seeking structure after coarse segmentation. The CNN structure combines three architectural ideas to ensure some degree of shift, scale, and distortion invariance, i.e., local receptive fields, shared weights, and pooling operations [18].

C. Pretraining Strategy

As described in the previous section, kernel w_i in (1) can be trained by a BP neural network; however, this can be very time consuming. For example, it took about 15 h to train kernel w_i in the previous CNN-based VLR system with 10 000 training logo images, which is not appropriate for a real-world application that requires frequently updated training data. The motivation for the strategy presented here is that kernel training in previous classical CNNs started from a random initialization, and we hope that the iterations used to update the kernel training procedure can be accelerated using a good initial value obtained by pretraining. In addition, unsupervised pretraining has been adapted in other studies using deeply layered architectures, where each layer is pretrained with an unsupervised learning algorithm [20], [21]. This kind of approach, where greedy layerwise unsupervised pretraining is followed by supervised fine tuning, has been reported to enhance the performance of deep neural networks since the parameter values for discriminant training can be set in the appropriate range by regularization from unsupervised pretraining. In addition, pretraining initializes the model to a point in the parameter space that somehow renders the optimization process more effective, in the sense of achieving a lower minimum of the empirical cost function [22].

Each convolutional layer in the CNN is implemented as follows: there are M training logo images, and the ith feature maps in the jth training image is defined as

$$y_j^i = w_i \otimes x_j \tag{4}$$

where $\otimes$ denotes the 2-D convolution, $w_i \in R^{l_x \times l_y}$ is the ith kernel, and $x_j \in R^{s_x \times s_y}$ denotes the jth training image; the boundary of x_j is zero padded before the convolution to make $y_j^i \in R^{s_x \times s_y}$ have the same size of x_j. Let $x_j^v \in R^{s_x s_y \times 1}$ be an image x_j in a vectorized form, and let P_l be the lth patch extraction matrix, i.e., P_l is an $l_x l_y \times s_x s_y$ matrix of all zeros, except for one in each row that extracts a vectorized $l_x \times l_y$ patch from image x_j^v, $x_j^l = P_l x_j^v \in R^{l_x l_y \times 1}$ for $l = 1, 2, \ldots, s_x s_y$. Therefore, (4) can be rewritten as

$$Y_j^i = W_i^T X_j \tag{5}$$

where $Y_j^i \in R^{1 \times s_x s_y}$ is the vectorized form of y_j^i, $W_i \in R^{l_x l_y \times 1}$ is the vectorized form of w_i, and $X_j = [x_j^1, x_j^2, \ldots, x_j^{s_x s_y}] \in R^{l_x l_y \times s_x s_y}$.

Therefore, the problem of identifying a good initial value for $\{w_i\}_{i=1}^N$ turns into the problem of seeking for a good initial value of $\{W_i\}_{i=1}^N$ by a pretraining procedure. Since there is no label information in this step, the pretraining step is unsupervised. By combining all the training images together, a new matrix is defined as $X = [X_1, X_2, \ldots, X_j, \ldots, X_M] \in R^{l_x l_y \times M s_x s_y}$. The pretraining aims to find a set of kernels/basis $\{W_i\}_{i=1}^N$ that is able to reconstruct X with a minimum reconstruction error. The first kernel W_1 can be solved as

$$\arg\min_{W_1} \left\| X - W_1 W_1^T X \right\|_F^2, \quad \text{s.t.} \quad W_1^T W_1 = 1. \tag{6}$$

The first kernel W_1 is optimized to minimize the reconstruction error of X, i.e., projection $W_1^T X$ should denote the maximum power of X, and $W_1^T W_1 = 1$ is the normalization. In order to simplify the problem, all kernels are assumed to be

orthogonal; thus the remaining $N-1$ kernels are iteratively calculated as

$$X^K = X - \sum_{k=1}^{K-1} W_k W_k^T X \tag{7}$$

$$W_K = \arg\min_{W_K} \left\| X^K - W_K W_K^T X^K \right\|_F^2$$

$$\text{s.t.} \quad W_r^T W_t = \begin{cases} 0, & \text{if } r \neq t \\ 1, & \text{if } r = s \end{cases} \tag{8}$$

where X^k is the updated X in the kth iteration, and $X_0 = X$. As described in (7), the sum of projections at X based on the first $K-1$ projections is subtracted from X. The minimum function in (8) aims to find the kernel W_K that can reconstruct the updated X^K with a minimum reconstruction error in the Kth iteration. The constraint in (8) denotes that all the kernels must be orthogonal to each other. The procedure used in (7) and (8) happens to be the same as PCA, which is more efficient in the implementation than the BP neural networks in the CNN model. Therefore, the solutions of (6) and (8) are the first N eigenvectors of matrix XX^T. It should be emphasized that the number of kernels N can be adaptively determined using a simple thresholding operation. If the power ratio of the first N_p components to X is larger than a given threshold, N_p is defined as the number of feature maps in the convolutional layer.

As previously described, our initial motivation is to find a good initial value to accelerate weight training for a real-world application, and this can be implemented with PCA very efficiently. However, we were very surprised that the pretraining weights were able to directly extract features that were strong enough to produce satisfactory classification results in the recognition procedure described in the next section, i.e., the convolutional layers in (1) can be directly implemented with the pretraining strategy from pretraining without any following training (BP).

III. Experiments

A. Data Set Descriptions

We generated a data set to assess the proposed system. The data set includes vehicle logos from top ten popular manufacturers, which comprises more than 80% of vehicle brands in mainland China. One thousand outdoor vehicle frontal area images were captured from cameras located below traffic lights in a local traffic monitoring system. As shown in the example in Fig. 5, only one or two cars exist in each image, and there is no overlapping or corruption on the logo image. The cars in the images can be then categorized into ten classes according to their manufacturers.

The license plate has been already detected and segmented using the recognition kit embedded in the traffic monitoring system. The proposed scheme has used coarse segmentation to detect the region containing the vehicle logo, which was normalized to 70×70 pixels in relation to the vehicle license plate. The original images from the monitoring system originate from various outdoor imaging situations. To further emphasize the robustness of the proposed scheme, the images were additionally treated with various distortions (illuminations, rotations, and different kinds of noise) to simulate various poor outdoor imaging situations, e.g., viewpoint variations, serious illumination variance, inaccurate license plate detection, and noise from the imaging recorder.

Fig. 5. Original image captured from the traffic monitor system.

Fig. 6. Some examples in the data sets. Ten manufacturers are included in our data sets. Various rotations, illuminations, and noise are also added on the original images from the imaging system to simulate various imaging procedures.

Finally, a vehicle logo data set with 11 500 images from 10 manufacturers was generated, where each manufacturer has 1150 images. Some sample images from the ten manufacturers in the data set are shown in Fig. 6. All the segmented images are normalized to 70×70 pixels. The data set is available at the project page http://smartdsp.xmu.edu.cn/VehicleLogoRecognition.html.

B. Experimental Results

The data set described earlier has been employed to validate the proposed system. Ten thousand of the images were used as training samples, and 1500 were used as testing samples. For each manufacturer, 1000 images were training samples, and 150 images were testing samples.

In the CNN-based VLR system, the empirically specified parameters of each layer in the CNN are presented in Table I. The original segmented image corresponds to layer 0. Six

TABLE I
SPECIFIED PARAMETERS OF THE MULTILAYER STRUCTURE IN THE PROPOSED CNN STRUCTURE

Layer	Type	Feature Maps	Kernel size
0	Input	1 map	-
1	Convolution	6 maps	7×7
2	Pooling	6 maps	2×2
3	Convolutional	12 maps	21×21
4	Pooling	12 maps	2×2
5	Full connection	-	-
6	Classifier	-	-

TABLE II
CLASSIFICATION ACCURACY ACROSS DIFFERENT VEHICLE MANUFACTURERS BY THE CNN

Manufacturer	True	False	Accuracy
Honda	147	3	98.00%
Peugeot	149	1	99.33%
Buick	149	1	99.33%
VW	150	0	100.00%
Toyota	132	18	88.00%
Lexus	150	0	100.00%
Mazda	148	2	98.67%
Chery	150	0	100.00%
Hyundai	147	3	98.00%
Citroen	150	0	100.00%
Average	1472/1500	28/1500	98.13%

convolutional kernels (7×7 for each kernel) lead to six feature maps in layer 1. Max pooling in layer 2 downsamples the feature maps in the previous layer with local maximum detection. Layers 3 and 4 are the same as layers 1 and 2, except for the kernel size in the convolution. Extracted local features are fully connected in layer 5. The BP neural network classifier in the last layer of the CNN is trained with the high order feature vectors of the training samples, and each query image can be categorized with its feature vector. The number of feature maps is a critical parameter in the CNN and related models. In the CNN, the kernel number is selected with optimized performance through multiple experiments. In the pretrained CNN, since PCA-based pretraining seeks for the projections of the signals on N_p, different orthogonal basis (weights) and, then, the power ratio of the first N_p components-to-original signal can be calculated as R_{N_p}. Therefore, the number of feature maps can be determined by simple thresholding, i.e., $R_{N_p} \geq T$, where T is the given threshold. In the proposed scheme, T is defined as 95%, giving an $N_p = 8$ in each convolutional layer.

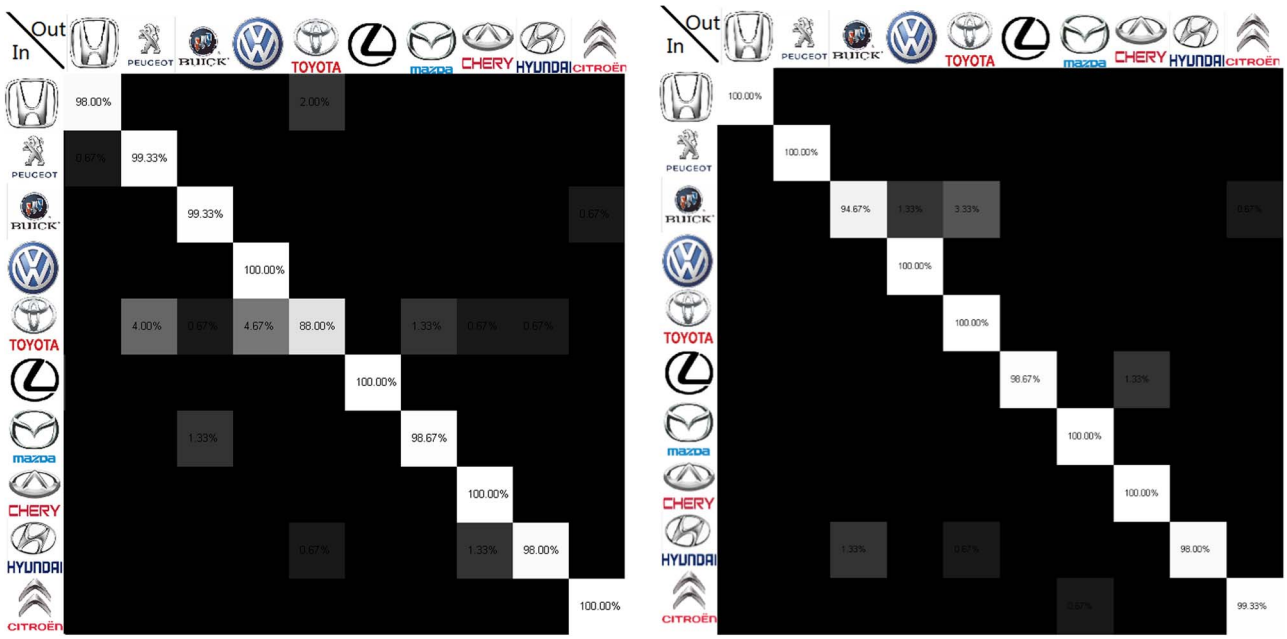
In the pretraining CNN-based VLR system, two simple and well-known supervised classifiers, i.e., the SVM and the BP neural network (with one hidden layer), are employed for the final recognition after feature extraction. Since BP has similar performance to the SVM but with a higher computational cost in the classifier training, the recognition results with the SVM are demonstrated as follows. All the experiments are implemented in Matlab 2012 on a personal computer with an Intel-I3 central processing unit and a 2-GB memory without optimization. The classification accuracy for the ten manufacturers from both the CNN and the pretraining CNN is summarized in Tables II and III. The accuracy is based on the assumption that LPL has 100% accuracy since coarse segmentation detects a region that is much larger than the logo. Fig. 7(a) and (b) illustrates the confusion matrix of the verification process from the two proposed systems. The main diagonal displays the high recognition accuracy of the proposed method. It can be observed that the pretraining CNN model slightly outperforms the CNN with less classification error.

TABLE III
CLASSIFICATION ACCURACY ACROSS DIFFERENT VEHICLE MANUFACTURERS BY THE PRETRAINING CNN

Manufacturer	True	False	Accuracy
Honda	150	0	100.00%
Peugeot	150	0	100.00%
Buick	142	8	94.67%
VW	150	0	100.00%
Toyota	150	0	100.00%
Lexus	150	0	100.00%
Mazda	148	2	98.67%
Chery	150	0	100.00%
Hyundai	147	3	98.00%
Citroen	149	1	99.33%
Average	1486/1500	14/1000	99.07%

According to Table III, there is a total of 14 wrongly recognized images with the pretraining CNN belonging to three categories (Buick, Mazda, and Hyundai). As shown in the examples in Fig. 8, it can be observed that the wrongly recognized images are with complex structures (Buick), are heavily blurred (Buick and Hyundai), and are with extreme uneven illumination, e.g., it is still a challenge to recognize the last two logo images in Fig. 8 manually. In addition, another tenfold cross validation is additionally implemented with the whole data set, with the recognition accuracy equal to $99.13\% \pm 0.24\%$.

The computational cost comparison between the two proposed methods is also provided in Table IV. The testing procedures in both proposed models are efficient enough for real-time implementation. The low computational cost is due to: 1) logo detection via coarse segmentation is very efficient since the assumption is very simple; and 2) the repeat max-pooling layer in the CNN structure downsamples the feature maps iteratively. The CNN is more efficient in the testing procedures than the pretraining CNN (12 versus 160 ms). It can be explained as follows: The testing procedures in both the CNN and the pretraining CNN contain feature extraction and classification. All the parameters in both proposed models are adaptively or empirically estimated for their optical performance. In both models, the classification step is very efficient; thus, almost all the differences in the computational cost are due to the feature extraction procedure ($C1$–$S4$ in Fig. 4). For more details, the parameter settings in layer $C3$ are quite different between the CNN and the pretraining CNN, e.g., the kernel sizes in the two models are 2121 and 77, respectively, which leads to a large difference in the computational costs, and the feature vector sizes in the two models after a fully connected layer (25 921 versus 207 361). More information on the computational cost analysis can be found in the project page. However, it should be emphasized that the computational cost of the training procedure of the CNN is much higher than that in the improved model (15 h versus 15 min). These results demonstrated that the pretraining CNN is more appropriate for real-world applications, where training procedures are repeating due to updated training samples.



(a) (b)

Fig. 7. Confusion matrix across different vehicle manufactures with (a) CNN and (b) pretraining CNN.

Fig. 8. Examples of wrongly recognized logo images in the pretraining CNN.

TABLE IV
COMPUTATIONAL COST COMPARISONS

Method	Training procedures	Testing procedures
CNN	15hr	12ms/image
Pre-training CNN	15min	160ms/image

TABLE V
ACCURACY COMPARISONS ON ANOTHER DATA SET

Methods	Logo recognition accuracy	Fully automatical
MFM[8]	94%	×
M-SIFT[19]	94.6%	×
CNN	**96.08%**	√
Pre-training CNN	**100%**	√

C. Accuracy Comparisons

For a fair comparison, both proposed models are validated on another public data set. We extended the segmented logo data set in [8], [19], and [23], and we applied our method on them. As in Table V, it can be observed that the proposed models outperform the reported method in another data set. It should be also noted that the methods in [8] and [19] in comparison are not fully automatic due to a manual reference image selection step; however, the proposed models are fully automatic, which is more appropriate for real-world applications. In addition, to our knowledge, the logo recognition accuracy of the proposed schemes is the highest logo recognition accuracy across different reports, and the proposed data set is the most complex and has the largest size [6], [8], [9], [19].

D. Discussions

As described earlier, additional distortions in the data set images were produced to generate images that are of poorer quality than those of a real public traffic. In this section, the robustness of the two proposed schemes will be validated against various rotations, noise, shifting, and even combined distortions, in order to simulate different image monitoring situations. The distortion parameters are estimated by experts in the traffic security department, and some images have a higher distortion level than those of the proposed data set. The purpose is to make sure that the distorted images in the validation data set have satisfied the most extreme cases that could be encountered in real-world applications.

1) Robustness to Rotations: The robustness of the proposed method to rotations was validated as follows. For quantitative analysis, 608 logo images from the 10 manufacturers without any rotation were first selected from the testing samples and then rotated within -12° to 12° as testing samples. The examples of logo images in Fig. 9 with different rotation degrees have demonstrated that the proposed range covers most rotation cases in different imaging situations. The average classification accuracy across all the testing images with various rotations is presented in Fig. 9. It can be observed that the accuracy of both models slightly decreases when the rotation degree increases, which demonstrates that both proposed models are robust to various rotations. The pretraining CNN has better performance than the CNN in most cases.

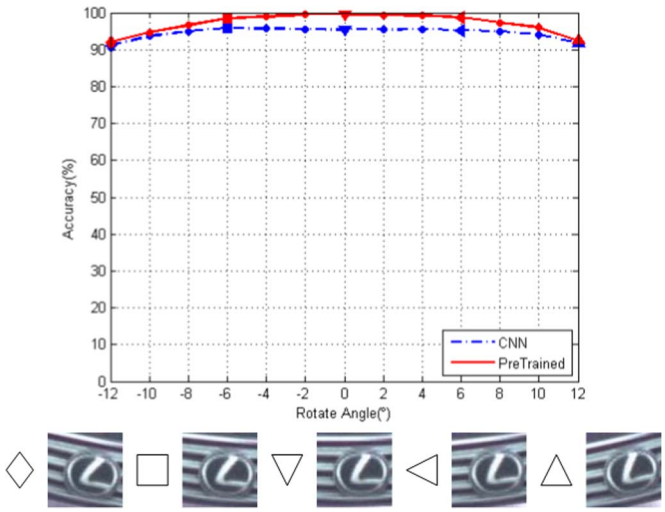


Fig. 9. Robustness against rotations.

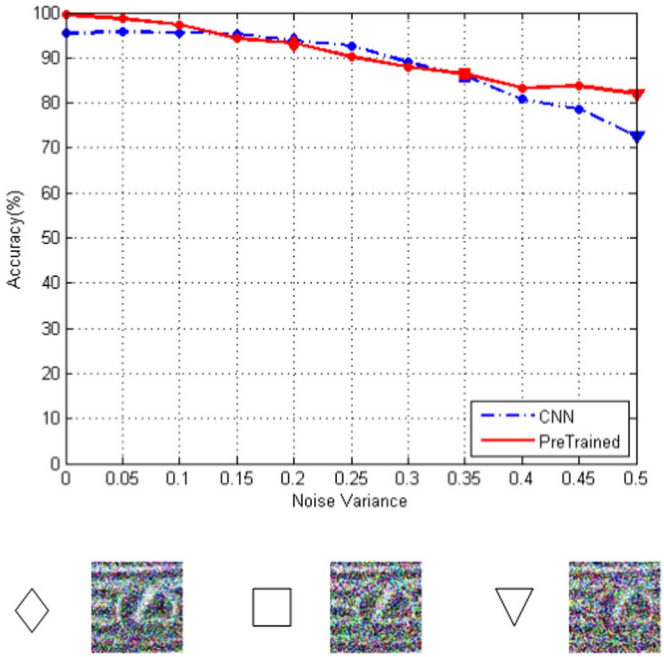


Fig. 10. Robustness against noise.

2) Robustness to Noise: Similar to the previous section, 608 logo images were randomly selected in the testing data set. For validation, Gaussian noise with different variances was added to the logo images. Examples of some noisy logo images corresponding to the different indicators are shown in Fig. 10. A plot of average classification accuracy with different noise variances across all the testing samples is presented in Fig. 10. The accuracy slightly decreases when the noise variance increases. The accuracy is above 75% and 81% for the CNN and the pretraining CNN, respectively, if the noise variance is 0.5 (with an indicator triangle). It can be observed that the image is hardly recognizable, even by observers.

3) Robustness to Shifting: After coarse segmentation, the logo can be located within any part of the segmented region, as shown in the examples in Fig. 6. The robustness against this "shifting" is also validated as follows. First, examples of nine

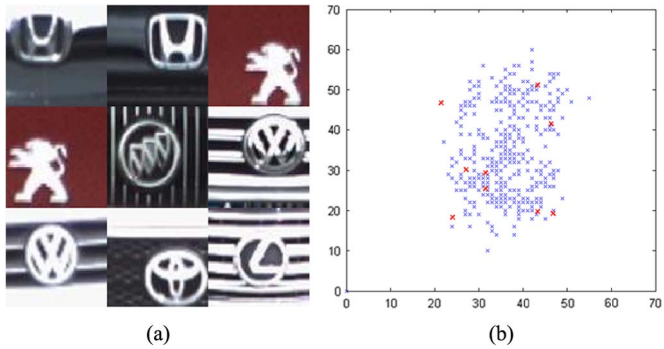


Fig. 11. Robustness to shifting. (a) Examples of successfully recognized logo images. (b) Histogram of logo location centers. The red crosses correspond to the example images at the bottom.

successfully recognized logo images from different manufacturers are presented in Fig. 11(a), which demonstrates that the successfully recognized logos can be located in different parts of the image such as top left, top middle, top right, etc. Second, for detailed analysis, the locations of 1500 testing logo images are also summarized in the histogram in Fig. 11(b). Each star indicates the central coordinate of the logo in each image. A proposed accuracy of 99.07%, along with the histogram, has demonstrated that the proposed system is robust to different logo locations based on simple coarse segmentation.

4) Robustness to Various Illuminations: Due to different imaging conditions, the robustness of the proposed model against various illuminations was also validated. Six hundred eight logo images from ten manufacturers were randomly selected, and red–green–blue (RGB)-to-hue–saturation–intensity (HSI) conversion was then applied to each image in order to determine the intensity component I of the image [24]. Various illumination simulations can be implemented by a gamma correction on I as

$$I_1 = 255 \times (I/255)^{\frac{1}{\gamma}} \tag{9}$$

where gamma is the scalar within the range (0.1, 50). HSI-to-RGB conversion is then applied to generate the "adjusted" images. The plots of the average recognition accuracy across 608 samples with different scalars (in longitude) are shown in Fig. 12. Sample logo images with different gamma values, corresponding to the indicators "diamond," "triangle," and "box" in the plot, are also shown. It can be observed that the accuracy always remains above 90% and 97% for the proposed CNN and the pretraining CNN, respectively, during various illumination levels.

5) Robustness to Various Combined Distortions: An experiment is implemented to validate the robustness with an online setting. A set of image samples (100 images) is generated with all the tested combined distortions (combinations of various noise variances, illuminations, and rotations), and then, two image subsets A and B are generated with sizes of 50 and 50, respectively. The experiment includes three recognition steps to simulate the real-world applications where the training samples are iteratively updated by testing samples after recognition, and a small percentage of the testing logo images is with extreme distortions. We also make sure that there is no overlap between

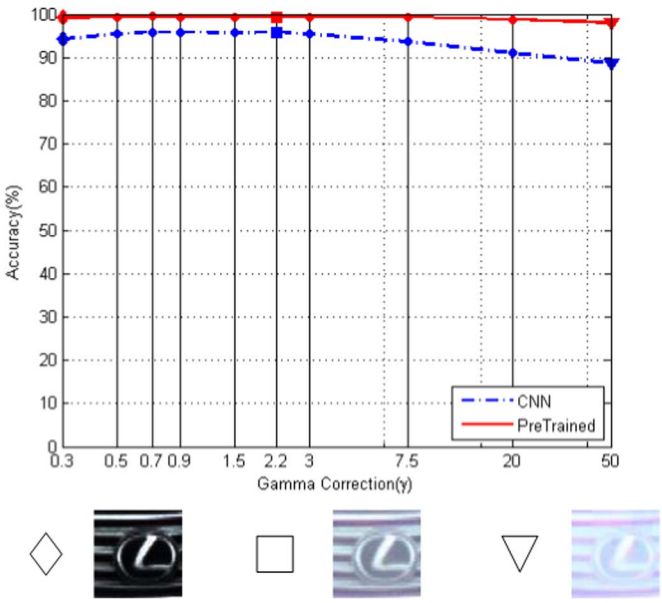


Fig. 12. Robustness validation against illuminations with different gamma values in (3), and the gamma is set as (0.3, 0.5, 0.7, 0.9, 1.5, 2.2, 3, 7.5, 20, 50).

the training samples and the testing samples in the experiment. The testing samples in Section III are randomly divided into three individual subsets, i.e., subsets C, D, and E, with sizes of 500, 450, and 450, respectively. Three examples of combined distorted logo images are shown in Fig. 13.

Step 1 The training samples are the same as in Section III (10 000 images). The testing samples are 500 images in subset C. There are no combined distorted images in the training samples or in the testing samples.

Step 2 We randomly abandon 500 images in the previous training samples. In addition, the previous testing samples in subset C are included in the updated training data with their correct labels; thus, the size of the training data is the same as before. Meanwhile, subset D is combined with subset A, generating testing data with a size of 500. There is no combined distorted image in the training data, but there are 50 combined distorted images in the testing data.

Step 3 Similar as Step 2, the training data are updated by replacing 500 randomly selected images with the testing data in Step 2. In addition, subset B combined with subset E is employed as the testing data. Thus, combined distorted images are included in both the training data and the testing data.

The accuracy of the three experiments are provided in Table VI. Accuracy 1, Accuracy 2, and Accuracy 3 denote the classification results of the total testing samples, the non-combined distorted data, and the combined distorted data, respectively. It can be observed that Accuracy 1 and Accuracy 2 are almost the same across the three steps. Accuracy 3 slightly decreases in Step 2 since there are no similar samples in the training data; however, it increases to 98% in Step 3 with the updated training data. The validation with an online setting has

Fig. 13. Three examples of combined distorted images in robustness validations.

TABLE VI
ACCURACY SUMMARIZATION IN ROBUSTNESS VALIDATION WITH COMBINED DISTORTED DATA

	Accuracy1	Accuracy2	Accuracy3
Step1	99.4%	99.4%	-
Step2	98.8%	99.3%	94%
Step3	99.6%	99.8%	98%

demonstrated the robustness of the proposed model on various combined distorted images.

In addition, another similar experiment is provided in the project page in order to simulate the extreme cases, where all the testing samples are with combined distortions in each step. The high accuracy also demonstrated the strong robustness of the proposed pretraining CNN model.

6) Further Discussions: In this paper, the vehicle logos from the top ten popular manufacturers in mainland China were used for logo recognition validation, and the area of coarse segmentation was defined as $m/2 \times m/2$ above the license plate, as per the analysis in Fig. 2. In order to extend this method to other applications, we must be cognizant of the fact that the distance between the license plate and the logo may vary across cars, such as SUVs and other vehicles. This can be easily addressed by using a larger coarse-segmented area to guarantee that the logo will be within the detected area. For example, the coarse segmentation area can be defined as $m \times m$ or even larger. Considering that the CNN can be implemented in parallel on Graphics Processing Unit (GPU) with a Compute Unified Device Architecture, it is still possible for the proposed scheme to be suitable for real-time applications.

IV. CONCLUSION

A VMR method based on a CNN model has been proposed in this paper. The proposed method removes the requirement for precise VLD and segmentation. The stacked trainable structure has the ability to extract features that will be robust against various translations, rotations, and noise variances. Due to the fact that the training samples are always updated in the application, a PCA-based pretraining strategy is introduced to improve the CNN-based system by reducing the computational cost of the training stages and enhancing the classification performance simultaneously. Both proposed systems have been evaluated on a data set containing the top ten popular vehicle manufacturers in mainland China. Experimental results have demonstrated that the proposed improved CNN is more appropriate for real-world applications. In addition, the CNN structure makes it suitable for parallel implementation in GPU, thus making a real-time recognition system possible.

REFERENCES

[1] D. Zheng, Y. Zhao, and J. Wang, "An efficient method of license plate location," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2431–2438, Nov. 2005.

[2] B. Hongliang and L. Changping, "A hybrid license plate extraction method based on edge statistics and morphology," in *Proc. 17th ICPR*, 2004, vol. 2, pp. 831–834.

[3] V. S. Petrovic and T. F. Cootes, "Analysis of features for rigid structure vehicle type recognition," in *Proc. BMVC*, 2004, pp. 1–10.

[4] L. Figueiredo, I. Jesus, J. Machado, J. Ferreira, and J. M. de Carvalho, "Towards the development of intelligent transportation systems," in *Proc. Intell. Transp. Syst.*, 2001, vol. 88, pp. 1206–1211.

[5] Y. Wang, Z. Liu, and F. Xiao, "A fast coarse to fine vehicle logo detection and recognition method," in *Proc. IEEE Int. Conf. Robot. Biomim.*, 2007, pp. 691–696.

[6] K. T. Sam and X. L. Tian, "Vehicle logo recognition using modest adaboost and radial tchebichef moments," in *Proc. 4th Int. Conf. Mach. Learn. Comput.*, 2012, pp. 91–95.

[7] L. Dlagnekov and S. Belongie, "Recognizing cars," Dept. Comput. Sci. Eng., Univ. California, San Diego, CA, USA, 2005.

[8] A. P. Psyllos, C. N. Anagnostopoulos, and E. Kayafas, "Vehicle logo recognition using a sift-based enhanced matching scheme," *IEEE Trans. Intell. Transp. Syst.*, vol. vol. 11, no. 2, pp. 322–328, Jun. 2010.

[9] S. Yu, S. Zheng, H. Yang, and L. Liang, "Vehicle logo recognition based on Bag-of-Words," in *Proc. 10th IEEE Int. Conf. AVSS*, 2013, pp. 353–358.

[10] S. Dai, H. He, Z. Gao, K. Li, and S. Xiao, "Vehicle-logo recognition method based on Tchebichef moment invariants and SVM," in *Proc. WCSE*, May 2009, vol. 3, pp. 18–21.

[11] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurons in the cat's striate cortex," *J. Physiol.*, vol. 148, no. 3, pp. 574–591, Oct. 1959.

[12] A. Ahmed, K. Yu, W. Xu, Y. Gong, and E. Xing, "Training hierarchical feedforward visual recognition models using transfer learning from pseudo-tasks," in *Proc. Comput. Vis. CECCV*, 2008, pp. 69–82.

[13] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.

[14] Y. Le Cun *et al.*, "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.

[15] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, " Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, vol. 35, no. 8, pp. 1915–1929 Aug. 2013.

[16] P. Sermanet and Y. Lecun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 2809–2813

[17] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[18] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 10, pp. 1610–1623, Oct. 2010.

[19] A. P. Psyllos, C. N. Anagnostopoulos, and E. Kayafas, "M-SIFT: A new method for vehicle logo recognition," in *Proc. IEEE Int. Conf. Veh. Electron. Safety*, 2013, pp. 24–27

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

[21] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. NIPS*, 2007, pp. 153–160.

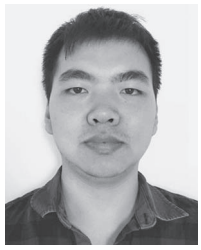
[22] D. Erhan *et al.*, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Mar. 2010.

[23] Images Database, Feb. 2009. [Online]. Available: http://www.medialab.ntua.gr/research/LPRdatabase.html

[24] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

Yue Huang was born in Fujian, China, in 1983. She received the B.S. degree from Xiamen University, Xiamen, China, in 2005 and the Ph.D. degree from Tsinghua University, Beijing, China, in 2010.

Since 2010, she has been an Assistant Professor with the Department of Communication Engineering, School of Information Science and Engineering, Xiamen University. Her main research interests include sparse signal representation and machine learning.

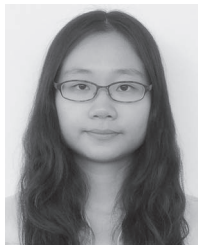


Ruiwen Wu received the B.S. degree from Xiamen University, Xiamen, China, in 2013. He is currently working toward the M.S. degree in the Department of Communication Engineering, School of Information Science and Engineering, Xiamen University.

His research interests include digital image processing, deep learning, and topic models.

Ye Sun received the B.S. degree from Jilin University, Changchun, China, in 2012. He is currently working toward the M.S. degree in the Department of Communication Engineering, School of Information Science and Engineering, Xiamen University, Xiamen, China.

His research interests include digital image processing and deep learning.



Wei Wang received the B.S. degree from Xiamen University, Xiamen, China, in 2013. She is currently working toward the Master's degree in the Department of Electronic Engineering, School of Information Science and Engineering, Xiamen University.

Her research interests include digital image processing, deep learning, and topic models.

Xinghao Ding (M'99) was born in Hefei, China, in 1977. He received the B.S. and Ph.D. degrees from Hefei University of Technology, Hefei, China, in 1998 and 2003, respectively.

Since 2011, he has been a Professor with the Department of Communication Engineering, School of Information Science and Engineering, Xiamen University, Xiamen, China. From September 2009 to March 2011, he was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, Pratt School of Engineering, Duke University, Durham, NC, USA. His main research interests include image processing, sparse signal representation, and machine learning.