Vision-Based Occlusion Handling and Vehicle Classification for Traffic Surveillance Systems

Jianlong Chang, Lingfeng Wang, *Member, IEEE*, Gaofeng Meng, *Member, IEEE*,
Shiming Xiang, *Member, IEEE* and Chunhong Pan, *Member, IEEE*

Due to the factors such as visual occlusion, illumination change and pose variation, it is a challenging task to develop effective and efficient models for vehicle detection and classification in surveillance videos. Although plenty of existing related models have been proposed, many issues still need to be resolved. Typically, vehicle detection and classification methods should be vulnerable in complex environments. Moreover, in spite of many thoughtful attempts on adaptive appearance models to solve the occlusion problem, the corresponding approaches often suffer from high computational costs. This paper aims to address the above mentioned issues. By analyzing closures and convex hulls of vehicles, we propose a simple but effective recursive algorithm to segment vehicles involved in multiple-vehicle occlusions. Specifically, a deep convolutional neural network (CNN) model is constructed to capture high level features of images for classifying vehicles. Furthermore, a new pre-training strategy based on the sparse coding and auto-encoder is developed to pre-train CNNs. After pre-training, the proposed deep model yields a high performance with a limited labeled training samples.

***Index Terms*—Visual occlusion, recursive segmentation, vehicle classification, deep convolutional neural network.**

I. Introduction

Vision-based traffic surveillance, an indispensable part of Intelligent Transport System (ITS), has been widely studied over past few years. Many applications, including transportation planning, traffic operating and highway capacity analysis, are based on the vehicle detection and classification. There are two main challenging problems that still need to be well resolved [1]. One problem is the reliability and the instantaneity of the vehicle detection influenced by the illumination and the visual occlusion. The other is the accuracy of the vehicle classification related to the factors including pose, illumination and viewpoint of camera.

In the literature, there have been a number of studies on vehicle detection [2]–[6] and classification [7]–[11] over the past decade. Technically, two common steps are implemented in most vehicle detection models, *i.e.*, foreground extraction and vehicle segmentation from the foreground. Due to visual occlusions, the performance of these vehicle detection methods may be largely degraded in the complicated traffic environments. As for the vehicle classification, one essential problem is how to extract robust features to represent the images. Although many methods [12]–[15] have been presented for the task of vehicle classification, they often suffer from appearance variations of scenes and objects.

This paper focuses on the task of vehicle detection with visual occlusions as well as the task of vehicle classification. To tackle the occlusion problem, we develop an efficient model based on the techniques of **R**ecursive **S**egmentation and **C**onvex **H**ull (**RSCH**). Specifically, we assume that vehicles are convex regions in foregrounds. Under this assumption, a subset decomposition optimization model is derived to deal with the vehicle occlusion problem. For the task of vehicle classification, a deep convolutional network (CNN) model is represented to manage it. Since collecting the labeled vehicle images is a troublesome and time-consuming task, a novel pre-training stage is proposed to combat overfitting when the labeled data is limited. Specifically, the convolutional layers are initialized based on the sparse coding model and the fully connected layers are pre-trained via the auto-encoder model, respectively. For brief, we refer this pre-training strategy as **SCAE**: **S**parse **C**oding and **A**uto-**E**ncoder based method.

To sum up, the main contributions of this work are highlighted as follows:

- The RSCH method treats the connected regions as sets and utilizes a subset decomposition optimization for dealing with multiple occluded vehicles. As a result, the task of vehicle detection with visual occlusions can be formally converted into an optimization problem that could be solved efficiently by recursion.
- The sparse coding and the auto-encoder methods are employed to initialize the convolutional and the fully connected layers, respectively. Experimental results show that SCAE achieves excellent performance with only limited labeled data. Furthermore, beyond those complicated models with the support of the hardware system of large scale of GPUs, our model can be implemented on the ordinary computer based on CPUs only, which significantly extends the application of our method.

The reminder of this paper is structured as follows. Section II reviews the related work. Section III presents the details of the proposed RSCH method. Section IV describes the developed SCAE model. In Section V, the experimental results are reported with sufficient discussions. Conclusions are presented in the last section.

Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang and Chunhong Pan are with the Department of National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: {jianlong.chang, lfwang, gfmeng, smxiang, chpan}@nlpr.ia.ac.cn.

Jianlong Chang is also with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China.

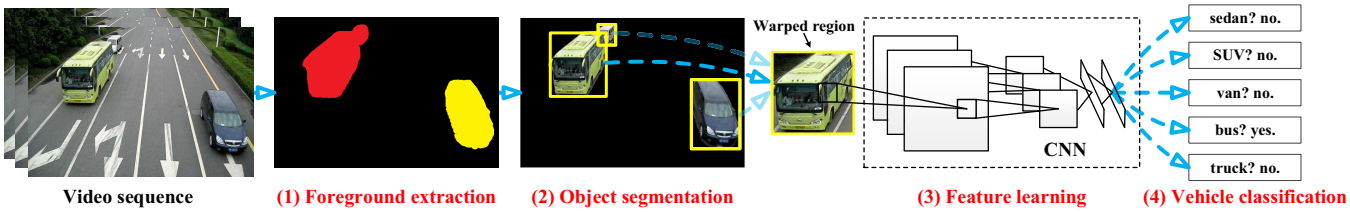


Fig. 1. Main flow of the proposed framework including four steps, *i.e.*, foreground extraction, object segmentation, feature learning and vehicle classification.

II. Related Work

Occlusion Problem: In prior studies, there are many solutions to address the vehicle occlusion problem. By calculating the direction of the contour and assigning a resolvability index to each occluded vehicle, Pang *et al.* [16] proposed a generalized deformable model (GDM) based method for vehicle segmentation in traffic images. While GDM can tackle occlusions effectively, the high computational complexity severely restricts the scalability in practical applications. Some real-time algorithms have been explored in [17], [18]. In [17], the occlusion is first detected by motion vectors of vehicles and then the occluded regions are segmented into two individual regions based on the "Cutting Region". This model can be treated as a clustering method based on motion vectors. However, it can only be used to solve the occlusion of two vehicles. Zhang *et al.* [18] proposed a multilevel framework to detect and handle vehicle occlusion, which tackles occlusion problem from the intraframe, interframe, and tracking aspects. With these different aspects, "cutting region" of the occluded vehicles, motion vectors of vehicles and occlusion layer images are utilized to handle occlusion. The results exhibit the effectiveness of the proposed framework. However, these methods have one main limitation: the occlusions of more than two vehicles are beyond their capabilities.

Vehicle Classification: In general, feature-based methods are commonly used for vehicle classification. Existing feature extraction methods can be grouped into two categories: the manually designed and the automatically extracted.

Among the former, Peng *et al.* [14] utilized scale-invariant feature transform (SIFT) descriptors to represent vehicle images. By leveraging sparse coding, the features are projected to a higher dimensional feature space. Then, a linear support vector machine (SVM) classifier is adopted to estimate vehicle types. In [15], the edge-based features and the modified SIFT descriptors were used to represent vehicle images. Moreover, Bayesian Decision rule is employed to distinguish each category of vehicles from others. While the SIFT feature descriptor can improve the effect of feature extraction, the local features of the images are described only and the global structure information can allow better performance for classification.

As a frequently used method for automatically learning features, deep learning is able to learn the informative features for vehicle classification [19]–[21]. Krause *et al.* [19] lifted two object representations from 2-dimensional to 3-dimensional to generalize across viewpoints based on a deep model. Based on a large-scale dataset, Yang *et al.* [20] presented a deep neural network model to distinguish fine-grained vehicle images. By using a pre-trained deep model, Zhou *et al.* [20] employed the fine-tuning strategy to improve the performance of vehicle classification on a specific large labeled dataset. Although such achievements are notable, two issues still require to be tackled. First, with a mass of parameters to be estimated, a large amount of labeled data is indispensable for training CNNs. However, collecting labeled data is troublesome and time-consuming. Second, high-performance servers are prerequisite to execute CNNs. Thus, with the purpose of extensive applications, effective CNNs with a small-size is required that can be implemented on CPUs only.

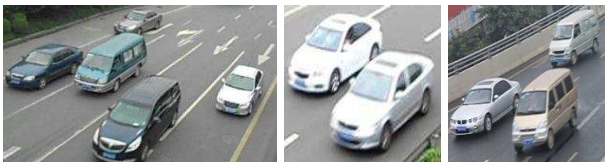


Fig. 2. Vehicle occlusion cases. (a,b) different number of vehicles, (b,c) different weather, (a,c) different camera angles.

III. Vehicle Detection

In this Section, we describe the proposed RSCH method. Specifically, the modified gaussian mixture model (GMM)-based model described in [18] is employed to extract the moving foreground since its robustness to illumination changes. After that, the probable existed occlusions are solved by the proposed RSCH method.

A. Motivation

There are two main reasons that give rise to occlusion: the limitation of camera angle and the heavy traffic. Ignoring the problem of occlusion may lead to erroneous estimates of traffic information. For example, the predicted number of vehicles will less than the actual value. As shown in Fig. 2, there are a variety of occlusion situations in practice, which indicates that a proper model must be common for all kinds of vehicle occlusion cases. According to the spatial relationships between vehicles, one should confront with three cases: no occlusion, partial occlusion and total occlusion. Following the previous

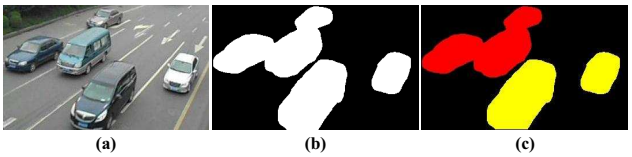


Fig. 3. The motivation of RSCH method. (a) Current frame, (b) foreground, (c) the yellow region represents total occlusion or no occlusion, and the red region means partial occlusion. In (c), when occur total occlusion or no occlusion, each connected region is a convex set or approximate to a convex set when total occlusion or no occlusion occur. However, it will be very different when some vehicles with partial occlusion.

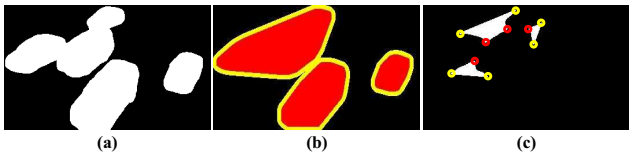


Fig. 4. Notations. (a) Connected region R^k. (b) The convex hull of R^k and $C(R^k)$ are depicted in yellow and red color, respectively. (c) Difference set D^k and $S(R^k)$. In (c), all Harris corners of D^k are depicted in colored circles. Specially, Harris corners which belong to the convex hull of R^k are marked in the yellow color circles, the rest in the red circles indicate $S(R^k)$.

work [16], we focus on the partial occlusions since the cases with no occlusion or total occlusion require no handling.

As shown in Fig. 3, each vehicle can be expressed by a convex set in foreground. Therefore, the occlusion problem can be considered as a problem of dividing the union of convex sets into several disjoint convex sets. In this paper, an optimal equation is established for this problem, and a recursive algorithm is proposed to solve it.

B. RSCH Model

In this section, we detail the proposed RSCH method.

1) Occlusion Detection

As shown in Fig. 4, for a connected region R^k, $C(R^k)$ represents the convex region which is enclosed by the convex hull of R^k. In the situations with no occlusion or total occlusion, R^k is close to $C(R^k)$; otherwise, R^k is much smaller than $C(R^k)$. Therefore, we introduce area difference function Γ to detect occlusions. Formally, Γ is defined as follows:

$$\Gamma(R^k) = card(C(R^k)) - card(R^k), \tag{1}$$

where $card(\cdot)$ is a function that returns the cardinality of a set. Based on $\Gamma(\cdot)$, the occlusion can be detected by:

$$Y = \begin{cases} 1, & \text{if } \Gamma(R^k) < \hat{\Gamma} \\ 0, & \text{if } \Gamma(R^k) \geq \hat{\Gamma} \end{cases}, \tag{2}$$

where $\hat{\Gamma}$ is an area difference threshold. For each region R^k and the area difference $\Gamma(R^k)$, $Y = 1$ indicates that R^k has partial occlusion; otherwise, R^k has no occlusion or total occlusion.

2) Problem Formulation

Once partial occlusion is detected in region R^k, the occlusion problem can be solved by finding a group of subsets $\{R_1^k, R_2^k, \cdots, R_N^k\}$ of R^k, in which only one vehicle is included in each subset. Accordingly, the objective of our model is to decrease the area difference $\Gamma(R_i^k)$. Formally, the final objective function can be formulated as follows:

$$\begin{aligned} &\min_{\{R_1^k, R_2^k, \cdots, R_N^k\}} \sum_{i=1}^{N} \Gamma(R_i^k) \\ &s.t.\ R^k = \bigcup_{i=1}^{N} R_i^k,\ R_i^k \bigcap R_j^k = \varnothing (i \neq j) \end{aligned}, \tag{3}$$

where R_i^k is a subset of R^k and N is the number of vehicles.

One major problem of the above model lies in the lack of appropriate constrained condition – it is possible to decrease the objective function to 0 by increasing the vehicles number N. To simplify the problem, we first consider the occlusion problem of two vehicles. Then, a recursive algorithm is proposed to manage more general cases.

Algorithm 1 RSCH-based method

Input: Foreground mask f, thresholds $\hat{\Gamma}$, ε_v and ε
Output: Vehicle set V

```
V = ∅
for each connected region R^k of f do
    Calculate C(R^k), D^k of R^k
    V ← V ∪ SEGMENT(R^k) //Algorithm 2
end for
```

3) Basic Segmentation

In fact, the segmentation problem can be treated as a clustering problem of 2D points. For partial occlusion caused by two vehicles, each vehicle is on different sides of a line. Therefore, it is reasonable to use a cutting line [17] to segment the occluded vehicles. Based on the cutting line, occluded regions can be segmented into two individual regions. Because it is costly to search for all possible of cutting lines, we narrow down the searching range of cutting lines to compress the time consumption as follows.

We detect corners of foregrounds by handling occlusion. Let D^k denote the different set between $C(R^k)$ and R^k. Since binary images have more defined corners than gray images, D^k is employ to detect occluded corners. These corners in convex hull are not occluded corners of vehicles. Therefore, the occluded corners are the different set between $H(D^k)$ and $CH(R^k)$, where $H(D^k)$ is a set of Harris corners [22] of D^k. In addition, $CH(R^k)$ represents the convex hull of $C(R^k)$. More formally, the occluded corners are detected as follows:

$$S(R^k) = H(D^k) - CH(R^k), \tag{4}$$

where $S(R^k)$ represents the occluded corners of R^k. Next, the points in the $S(R^k)$ are employed as the cutting points, and lines between pairs of cutting points are considered as the cutting lines. Since the difference of vehicle size is not very large when vehicles locate in similar position, it is reasonable

Algorithm 2 SEGMENT(R^k)

Input: A subset R^k
Output: Vehicle set in R^k.

1: Calculate $\Gamma(R^k) = card(C(R^k)) - card(R^k)$ // Eq. (1)
2: **if** $\Gamma(R^k) \geq \hat{\Gamma}$ **then**
3: **return** $\{R^k\}$
4: **else**
5: Calculate cutting point set $S(R^k)$ // Eq. (4)
6: Calculate T, which bring the maximum of Eq. (5)
7: Calculate $R^k_{i,1}, R^k_{i,2}$ using cutting line $line(P_{T[i]})$
8: $t = \arg\min_i |card(R^k_{i,1}) - card(R^k_{i,2})|$
9: $R^k_1 = R^k_{T[t],1}, R^k_2 = R^k_{T[t],2}$
10: Calculate $e(R^k) = \gamma(R^k) - \Gamma(R^k)$ // Eq. (6)
11: **if** $e(R^k) \leq \varepsilon$ **then**
12: **return** $\{R^k\}$
13: **else**
14: **return SEGMENT**$(R^k_1) \cup$ **SEGMENT**(R^k_2) // Eq. (7)
15: **end if**
16: **end if**

to constrain the segmentation based on the difference of vehicle size. Finally, we reform the optimal equation in Eq. (3):

$$\begin{aligned} &\min_t \ \Gamma(R^k_1) + \Gamma(R^k_2) \\ &s.t. \ R^k = r\big(R^k_1, R^k_2, line(P_t)\big) \quad , \\ &\quad |card(R^k_1) - card(R^k_2)| < \varepsilon_v \end{aligned} \tag{5}$$

where P is the Cartesian product of $S(R^k)$ and $S(R^k)$. In addition, $line(P_t)$ represents the cutting line between the two cutting points of the t^{th} element of P. The first constrained condition indicates that R^k is divided into R^k_1 and R^k_2 through $line(P_t)$. Specifically, a regularization term $|card(R^k_1) - card(R^k_2)| < \varepsilon_v$ is introduced to avoid over segmentation, where ε_v is a regularization threshold. In order to prevent the appearance of more than one optimal solution, RSCH chooses the cutting line which can minimize the cost of regularization term. To further reduce unnecessary segmentation, we introduce a variable e which indicates the difference of objective function in Eq. (5) between before and after segmentation. If there is a large difference, the segmentation is valid; otherwise, there is no point in dividing R^k to R^k_1 and R^k_2. For clarity, let $\gamma(R^k)$ denote the optimal value of Eq. (5), then e can be accordingly defined as follows:

$$e(R^k) = \gamma(R^k) - \Gamma(R^k). \tag{6}$$

When $e(R^k) > \varepsilon$, $\{R^k_1, R^k_2\}$ can be regarded as a reasonable segmentation scheme of R^k; otherwise, R^k is treated as a single vehicle without occlusion. Additionally, the constant ε is a threshold that is determined by experiment.

4) Recursive Segmentation

More generally, inspired by the recursive algorithm, a general occlusion problem can be solved by recursive segmentation as illustrated in Fig. 5, *i.e.*,

$$\textbf{SEGMENT}(R^k) = \textbf{SEGMENT}(R^k_1) \cup \textbf{SEGMENT}(R^k_2), \tag{7}$$

where **SEGMENT**$(\cdot)$ is a function which can divide a region into two parts, and $\{R^k_1, R^k_2\}$ is a segmentation scheme of R^k.

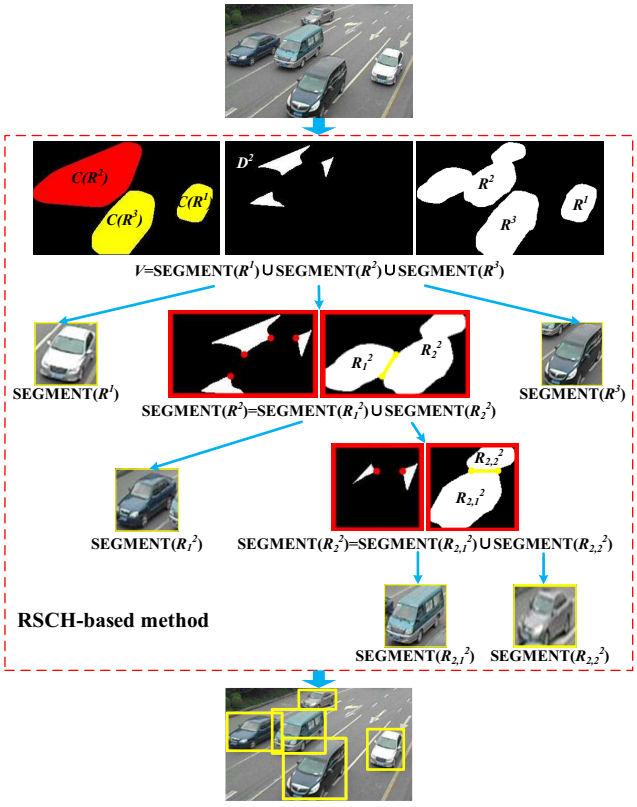


Fig. 5. Vehicle segmentation via the proposed RSCH method. For each connected region R, cutting point set $S(R)$ and the optimal cutting line are depicted in red and yellow color, respectively.

An illustration of the proposed RSCH method can be found in Algorithm 1. The inputs are the foreground mask f, the area difference threshold $\hat{\Gamma}$ which determines whether there is occlusion inside a connected region, and threshold ε which determines whether to make a segmentation for a connected region. By handling each connected region of f, vehicles are detected gradually. As described in Algorithm 2, for each connected region R^k, the segmentation function **SEGMENT**(R^k) is employed to segment vehicles of connected region. As described in the 5-15 lines, if R^k has partial occlusion and can be segmented as $\{R^k_1, R^k_2\}$, we will segment R^k_i $(i = 1, 2)$ recursively until they can not be segmented; otherwise, R^k is treated as a single vehicle.

IV. Vehicle Classification

In this section, SCAE model is developed to classify the vehicles detected by our RSCH method.

A. Motivation

A straightforward way to train an excellent deep neural network is increasing the number of labeled training data [23]. However, it is laborious and expensive to obtain a large amount of labeled data. Since training deep neural network can be regarded as an optimal problem with a group of parameters [24], it is reasonable to learn deep models by pre-training

these models with a mass of unlabeled data and fine-tuning the models for the specific classification task with labeled data.

In order to train a deep model with limited labeled data, a new pre-training strategy is proposed to find the initial point in this paper. The new pre-training strategy is motivated by some studies in biology. Hubel *et al.* have discovered the receptive field based on the researches of simple cells in mammalian primary visual cortex [25]. Furthermore, Olshausen *et al.* have shown that sparse coding algorithm can develop a group of complete receptive fields which are similar to Hubel's discovery in the visual cortex [26]. Inspired by that, a group of feature detectors learned by sparse coding are applied to extract low level features in our model. Furthermore, the fully connected layers are employed to yield more abstract features to represent images, as designed in [27]. More details are given in the following subsections.

B. Architecture of SCAE

Let $D_t = \{\mathbf{x}^k \in \mathbb{R}^{h\times w}\}_{k=1}^m$ denote m unlabeled images, where $\mathbf{x}^k$ is the k^{th} image in D_t, h and w represent the height and width of each image, respectively. We represent $D_f = \{(\mathbf{x}^k, \mathbf{y}^k)\}_{k=1}^n$ as a set of n labeled images with their labels, which are denoted by $\mathbf{y}^k \in \mathbb{R}^{c\times 1}$, and c indicates the number of vehicle categories. Specifically, $\mathbf{y}_p^k = 1$ and $\mathbf{y}_q^k = 0$ $(q \neq p)$ if $\mathbf{x}^k$ belongs to the p^{th} category.

1) Feature Extraction

There is essential difference between vehicle classification and large scale image classification. In total, the latter is more complicated and requires more deep architecture to make it. However, more deeper architectures mean that they require more labeled data to learn parameters and more powerful machine to execute models. Therefore, a shallow architecture is preferred for vehicle classification. As shown in Fig. 6, SCAE consists of two parts: convolutional feature learning and stacked auto-encoder feature learning.

The architecture of convolutional feature learning is similar to convolutional neural networks [28], and consisting of two layers: (1) a convolutional layer "C1", which is obtained as convolution of the input image and the learnable kernels, and (2) a pooling layer "S2", which following the convolutional layer "C1". For the image $\mathbf{x}^k$, its i^{th} feature map $\mathbf{x}_i^{k\text{-}conv}$ and pooling map $\mathbf{x}_i^{k\text{-}pool}$ are computed as follows:

$$\mathbf{x}_i^{k\text{-}conv} = ReLU(\mathbf{x}^k * \mathbf{F}^i + b_i), \tag{8}$$

$$\mathbf{x}_i^{k\text{-}pool} = Maxpooling(\mathbf{x}_i^{k\text{-}conv}), \tag{9}$$

where $\mathbf{F}^i \in \mathbf{R}^{d\times d}$ and b_i are the kernel weights with size $d \times d$ and basis, respectively. In addition, "$*$" denotes the convolution operation, $ReLU(\cdot)$ indicates the rectified linear function [29], *i.e.*,

$$ReLU(\mathbf{x}) = \max(\mathbf{0}, \mathbf{x}), \tag{10}$$

where $\mathbf{x}$ is the input of the $ReLU(\cdot)$ function. In addition, $Maxpooling(\cdot)$ represents max-pooling function. For each feature map, the max-pooling can obtain robust features, while losing some spatial information. To tackle this issue, we assemble the features of each pooling map to remedy the information loss, *i.e.*,

$$\mathbf{h}_0^k = \bigcup_i^N Vec(\mathbf{x}_i^{k\text{-}pool}), \tag{11}$$

where $Vec(\cdot)$ is a vectorization function to reshape matrixes to vectors, N represents the number of convolution kernels, $\bigcup$ indicates a function to concatenate a set of vectors to a vector and $\mathbf{h}_0^k$ represents the output of the convolutional feature learning part with the input $\mathbf{x}^k$.

We introduce the stacked auto-encoder feature learning part in our model to learn high-level features. In fact, it is not independent between different pooling maps. That is, they have a complicated interrelationship. Therefore, we assemble the pooling maps to obtain more complicated features, and employ stacked auto-encoder to learn the high level features of image. The functions in the stacked auto-encoder feature learning part can be formulated as follows:

$$\mathbf{h}_1^k = ReLU(\mathbf{W}^1\mathbf{h}_0^k + \mathbf{b}^1), \tag{12}$$

$$\mathbf{h}_2^k = ReLU(\mathbf{W}^2\mathbf{h}_1^k + \mathbf{b}^2), \tag{13}$$

where $\mathbf{W}^1$, $\mathbf{W}^2$ and $\mathbf{b}^1$, $\mathbf{b}^2$ are the weight matrices and bias of "L3" and "L4" layer, respectively. In addition, $\mathbf{h}_1^k$, $\mathbf{h}_2^k$ are the outputs of "L3" and "L4" layer, respectively.

2) Classification

As shown in Fig.6, softmax classifier is employed to classify images. We model the output by a probability vector $\mathbf{O}^k$, which can be formulated as follows:

$$\mathbf{O}^k = \frac{exp(\mathbf{W}^s\mathbf{h}_2^k)}{\sum_{i=1}^c \exp(\mathbf{W}_{i\cdot}^s\mathbf{h}_2^k)}, \tag{14}$$

where c indicates the number of vehicle categories, $\mathbf{W}^s$ is the projection matrix that maps $\mathbf{h}_2^k$ to output, and $\mathbf{W}_{i\cdot}^s$ denotes the i^{th} row vector of $\mathbf{W}^s$.

Algorithm 3 Training strategy of CNN

Input: unlabeled dataset D_t, labeled dataset D_f
Output: $\mathbf{W} = (\{\mathbf{F}^i\}_{i=1}^N, \mathbf{b}, \mathbf{W}^1, \mathbf{b}^1, \mathbf{W}^2, \mathbf{b}^2, \mathbf{W}^s)$

```
Random initialization ({F^i}_{i=1}^N, b, W^1, b^1, W^2, b^2, W^s);
// pre-training
Sample image batches X_b on D_t;
Update {F^i}_{i=1}^N based on X_b; // Eq. (15)
Calculate {h_0^k}_{k=1}^n by using {F^i}_{i=1}^N on D_t;
Update W^1, b^1 based on {h_0^k}_{k=1}^n; // Eq. (18)
Calculate {h_1^k}_{k=1}^n by using {F^i}_{i=1}^N, W^1, b^1;
Update W^2, b^2 based on {h_1^k}_{k=1}^n; // Eq. (18)
// fine-tuning
Update W based on D_f; // Eq. (19)
```

C. Training Strategy

The training process of our model is a two-step strategy. First, we adopt unsupervised methods to pre-train the devised CNN since the labeled data is limited. Second, the pre-trained network are fine-tuned with supervised methods.

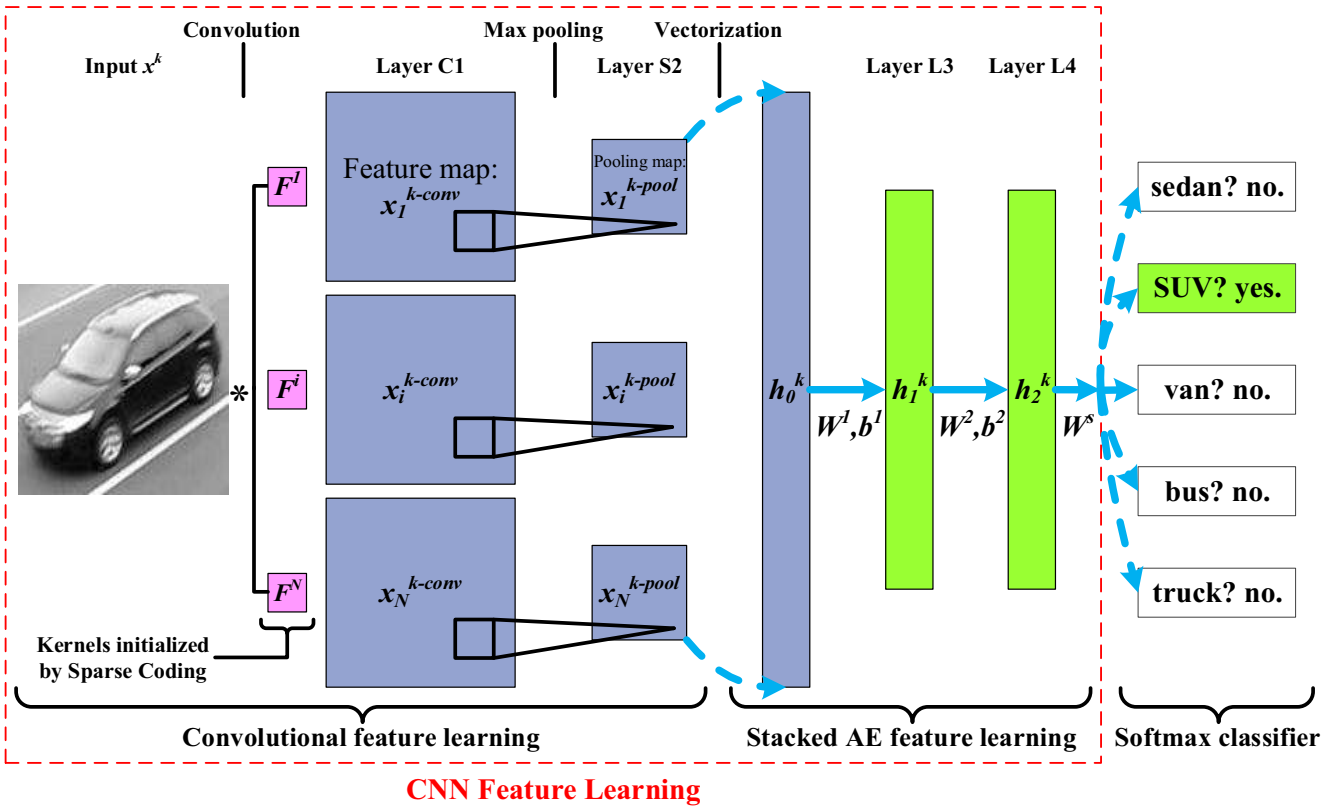


Fig. 6. The structure of SCAE. The input is a gray scale image. Through two specific feature learning steps, the visual features are mapped into a latent space and then a softmax classifier is employed to classify the image.

1) Unsupervised Pre-training Strategy

The proposed CNN is pre-trained on the unlabeled dataset D_t based on the sparse coding and the auto-encoder methods.

We attempt to learn a group of sparse feature detectors to initialize the convolution kernels $\{\mathbf{F}\}_{i=1}^N$. This is because sparse kernels can significantly improve the performance of classification tasks, as analyzed in [27]. Specifically, sparse coding attempts to find a dictionary matrix $\mathbf{B} \in \mathbf{R}^{d^2 \times N}$ to find sparse representations $\mathbf{Z} \in \mathbf{R}^{d^2 \times n_b}$ for unlabeled data sample $\mathbf{X}_b \in \mathbf{R}^{N \times n_b}$, where n_b is number of samples. Formally, the objective function is formulated as follows:

$$\ell_{SC}(\mathbf{X}_b; \mathbf{B}, \mathbf{Z}) = \sum_{i=1}^{N} \frac{1}{2}\|\mathbf{x}^i - \mathbf{B}\mathbf{z}^i\|_2^2 + \lambda\sqrt{\|\mathbf{z}^i\|_2^2 + \varepsilon} + \gamma\|\mathbf{B}\|_2^2, \quad (15)$$

where $\mathbf{x}^i$ is the i^{th} column vector in $\mathbf{X}_b$ and indicates a vector representation of image patch obtained by random sampling from the unlabeled dataset D_t. Moreover, $\mathbf{z}^i$ is the i^{th} row vector in $\mathbf{Z}$ and implies a sparse representation of $\mathbf{x}^i$. In Eq. (15), λ and γ are the regularisation parameters of the representation and basis vectors, respectively.

Eq. (15) can be optimized by alternately optimizing for $\mathbf{Z}$ for a fixed $\mathbf{B}$, and then optimizing for $\mathbf{B}$ given a fixed $\mathbf{Z}$. It turns out that this works quite well in practice. In this paper, $\lambda = 5 \times 10^{-5}$, $\gamma = 1 \times 10^{-2}$ and $\varepsilon = 1 \times 10^{-5}$. Conjugate gradient method are used to optimize the Eq. (15). Finally, the obtained N column vectors $\{\mathbf{b}^i \in \mathbf{R}^{d^2}\}$ of $\mathbf{B}$ are resized to $d \times d$ to initialize the convolution kernels $\{\mathbf{F}^i\}_{i=1}^N$.

We pre-train the stacked auto-encoder feature learning part layer by layer with each layer being an auto-encoder [30]. For clarity, let $\mathbf{v}$ denote the input of the auto-encoder. An auto-encoder is a two layer neural network defined as follows:

$$\mathbf{h} = ReLU(\mathbf{W}^0\mathbf{v} + \mathbf{b}^0), \quad (16)$$

$$\hat{\mathbf{v}} = ReLU(\hat{\mathbf{W}}^0\mathbf{h} + \hat{\mathbf{b}^0}), \quad (17)$$

where $\mathbf{h}$ indicates the latent representation of $\mathbf{v}$, $\hat{\mathbf{v}}$ is the reconstruction of the input $\mathbf{v}$ and $\mathbf{W}^0$, $\hat{\mathbf{W}}^0$ are the encoder and decoder parameters, respectively. Our goal is to minimize the reconstruction error with the following objective function:

$$\ell_{AE}(\mathbf{v}; \mathbf{W}^0, \mathbf{b}^0, \hat{\mathbf{W}}^0, \hat{\mathbf{b}^0}) = \frac{1}{2}\|\mathbf{v} - \hat{\mathbf{v}}\|_2^2. \quad (18)$$

In Eq. (18), the encoder parameters $\mathbf{w}^0$, $\mathbf{b}^0$ are employed to initialize the fully connected networks. Specifically, error backpropagation and RMSProp [31] optimizer are used to solve the Eq. (18). Technically, the "L3" layer is first pre-trained by reconstructing the output of the pre-trained convolutional feature learning part. Then, the "L4" layer is pre-trained based on the output of the "L3" layer.

In summary, we illustrate the pre-training strategy in Algorithm 3. The devised CNN is pre-trained in a layer-wise way. First, the convolution kernels are pre-trained based on

the sparse coding method. Second, the fully connected layers are pre-trained layer by layer via auto-encoder.

2) Fine-tuning Strategy

Except for last fully connected layer, the parameters of convolutional and stacked auto-encoder feature learning parts are initialized by aforementioned pre-training strategy. To enable the proposed CNN adapt to specific classification task, the entire CNN is fine-tuned in a supervised learning way. Given the labeled dataset, the objective function is formulated as follows:

$$\min_{\mathbf{W}} -\frac{1}{c}\sum_{k=1}^{m}\sum_{h=1}^{c}(\mathbf{y}_h^k \ln \mathbf{O}_h^k + (1-\mathbf{y}_h^k)\ln(1-\mathbf{O}_h^k)), \quad (19)$$

where $\mathbf{W} = (\{\mathbf{F}^i\}_{i=1}^N, \mathbf{b}, \mathbf{W}^1, \mathbf{b}^1, \mathbf{W}^2, \mathbf{b}^2, \mathbf{W}^s)$ represents entire parameters of network, and c is the number of vehicle categories. Error backpropagation and RMSProp optimizer are utilized to calculate the optimal value of the objective function.

V. Experimental results

In this section, we apply the proposed methods to detect and classify vehicles on surveillance videos. The algorithms are written in Visual C++ on a 2.5 GHz Core i5 PC with 4GB DDR3 RAM. The core code and the partial video data are shared at https://github.com/vector-1127/SCAE.

A. Vehicle Detection

1) Dataset

The test data consists of more than 5 hours of RGB scale videos, which were captured at the rate of 25 frames/s with an image size of 640×480 pixels on expressways in Chengdu during the day. The videos are taken from the heavy traffic environment, large amount of vehicles with visual occlusion, motion blur as well as different visual angles make detection more difficult.

2) Evaluation Protocol

Based on the same foreground detection model [18], we compared the performance of the RSCH method with other methods. We set $\hat{\Gamma} = 500$, $\varepsilon_v = 2000$ and $\varepsilon = 300$. Specifically, we split the occluded foreground images to occlusion samples of two and more than two vehicles to make comparisons, respectively. For two vehicles occlusion, CR [17] and MF [18] methods are employed to make comparisons. For more than two vehicles occlusion, GDM model [16] is employed to make comparisons. Furthermore, the F-measure and the average of time consumption are utilized to measure the stability of methods. The precision and the recall are calculated as follows:

$$\begin{aligned} P &= \frac{\text{The number of vehicles detected correctly}}{\text{The number of detected vehicles}}, \\ R &= \frac{\text{The number of vehicles detected correctly}}{\text{The number of vehicles in testing data}}, \end{aligned} \quad (20)$$

where P and R are represent the precision and recall, respectively. In addition, F-measure $F1$ can be calculated as follows:

$$F1 = \frac{2 \times P \times R}{P + R} \quad (21)$$

TABLE I
The results of different methods.

	two vehicles			more than two vehicles	
Methods	CR [17]	MF [18]	**Ours**	GDM [16]	**Ours**
P	0.94	0.95	0.97	0.96	0.95
R	0.87	0.92	0.93	0.92	0.90
F1	0.90	0.93	0.95	0.93	0.92
time	28ms	33ms	34ms	145ms	45ms

3) Results and Comparison

The test videos have 487 and 253 occluded images of two vehicles and more than two vehicles, respectively. The accuracy and the average time consumption of different methods, including CR [17], MF [18] and GDM [16], are summarized in Table I.

As shown in Table I, the proposed method is a compromise of accuracy and efficiency. For the occluded problem of two vehicles, the proposed method outperforms other methods. For the occlusions with more than two vehicles, in terms of accuracy, GDM [16] is better than RSCH method. The main reason is that the foreground utilized in [16] is more accurate. However, it is high time-complexity to calculate an accurate foreground. In summary, our RSCH method can be considered as a trade-off between efficiency and effect.

4) Discussion

The proposed RSCH method requires no more a prior knowledge of the vehicle other than an empirical assumption that vehicle is a convex region in foreground. Although this property depends on the accuracy of foreground detection and resolution of videos, it significantly avoids the influence of vehicles' shape and direction on vehicle detection. At the same time, low time-consumption extends its application in practice.

B. Vehicle Classification

1) Dataset

In this paper, the deep neural networks library Keras [32] is employed to train neural network. The 136726 vehicle images in the CompCars dataset [20] are employed as the unlabeled dataset D_t to pre-train the devised CNN. The labeled dataset D_f contains 5000 images, which were automatically collected from the aforementioned surveillance videos by using our proposed vehicle detection method. Among these samples, we sampled 4000 images to form the training dataset D_f^{train}, and the remainder 1000 images were then grouped into the testing dataset D_f^{test}.

2) Evaluation Protocol

We first select the hyper-parameters based on the labeled dataset, then we retrain the developed CNN with these settings according to our training strategy.

To train well our model (namely the CNN), the hyper-parameters are first selected in a way of the cross-validation. This task is conducted on the training dataset D_f^{train}. Specifically, 10-fold cross-validation is performed to achieve this

TABLE II
THE CONFIGURATION OF SUPERVISED METHODS.

AlexNet-like [27]	VGG-like [33]	**Ours**
conv3-64	conv3-64	conv11-25
maxpool2	conv3-64	maxpool4
conv3-128	maxpool2	FC-1024
maxpool2	conv3-128	FC-512
conv3-256	conv3-128	softmax
maxpool2	maxpool2	
FC-512	FC-512	
softmax	softmax	

TABLE III
THE RESULTS OF SEVERAL METHODS (%).

Methods	sedan	SUV	van	bus	truck	accuracy	time/image
AlexNet-like	95.5	94.8	95.7	95.0	96.5	95.50	47ms
VGG-like	94.8	94.4	94.9	95.3	96.5	95.18	58ms
SCAE^0	91.5	90.8	89.5	95.5	97.3	92.92	22ms
SCAE^1	89.2	88.8	87.2	92.7	94.3	90.44	**20ms**
SCAE^2	90.2	93.1	92.3	96.7	97.1	93.88	22ms
SCAE	**98.5**	**95.9**	**96.7**	**98.4**	**98.6**	**97.62**	22ms

goal, where the label information of the samples in D_f^{train} is also utilized. The design of the 10-fold cross-validation can be explained as follows. Specifically, we random assigned the samples in the dataset D_f^{train} into 10 subsets of equal size. This means that each group of the hyper-parameters with given values will be evaluated 10 times. In each time of parameter evaluation, nine subsets will be employed to train the model, and the rest one will be used to test the model.

In the training stage, the developed SCAE method with the unlabeled dataset D_t are utilized to pre-train our CNN model first. Then, the labeled dataset D_f^{train} and D_f^{test} are employed to fine-tune and test the devised CNN, respectively. In testing phase, given the trained CNN and a test image $\mathbf{x}$, the classification procedures are as below. First, we resize $\mathbf{x}$ into 64×64. Next, the output $\mathbf{O}$ is calculated through the forward pass of the entire network. In the end, label L is assigned to the image $\mathbf{x}$ as follows:

$$L = \arg\max_i O_i, 1 \leq i \leq c, \tag{22}$$

where c represents the number of categories, O_i is treated as the probability of $\mathbf{x}$ belong to the i^{th} category. For a reasonable evaluation, we perform 10 random restarts for all experiments and the average results are employed to compare with the others methods. Specifically, the method of data augmentation described in [27] is employed to combat overfitting. The form of data augmentation consists of randomly generating image rotations and translations only. Finally, the devised CNN model is trained again with these settings according to our proposed pre-training method. Error backpropagation and RMSProp optimizer are utilized to train the networks based on the different methods. The learning rate is 0.001 for the initial phase of training. The batch size is 128 during learning.

3) Compared Methods

Two baseline networks are devised in our work, namely, the AlexNet-like networks and VGG-like networks, which are similar to AlexNet [27] and VGG [33], respectively. There is no pre-training process when using AlexNet-like and VGG-like networks to test. The configuration of networks are listed as Table II. The convolutional layer parameters are denoted as "conv(receptive field size)-(number of channels)". The max-pooling layer parameters are denoted as "maxpool(pooling size)". The fully connected layer parameters are denoted as "FC-(output dim)". The ReLU activation function is employed as activation function because of its high efficiency and excellent effect [27]. For each network, the convolutional stride is fixed to 1 pixel, the spatial stride of max-pooling layer is equal to the pooling size, and the border mode of convolutional layer and pooling layer is "valid".

Furthermore, we compare the following additional methods to evaluate the effect of the SCAE pre-training strategy:

- SCAE^0: In contrast to SCAE, there is not pre-training process in SCAE^0.
- SCAE^1: In contrast to SCAE, SCAE^1 only contains the convolutional feature learning part.
- SCAE^2: In contrast to SCAE, only convolutional feature learning part is pre-trained in SCAE^2.

4) Classification Results and Comparison

The details of the classification results of aforementioned methods are shown in Table III. As shown in Table III, the comparative experiment results indicate that our method can achieve superior performance than other methods. Compared with the proposed CNN, AlexNet and VGG have more complicated structures. Therefore, when training data is limited, they will be more easy to cause overfitting and degrade performance. Furthermore, the developed CNN is high efficiency, which can extend the practicability of our method.

Compared with SCAE^0 and SCAE^2, SCAE has high capability for vehicle classification. This demonstrates that SCAE can initialize the good parameters for the devised CNN and improve the performance. In contrary to SCAE^1, SCAE can obtain more informative features by combining low-level features. The results also verify that the combination is effective.

5) Experiment for Pre-training Strategy

In order to further evaluate the effect of the developed pre-training strategy, some traditional state-of-the-art unsupervised methods including convolutional restricted Boltzmann machine (CRBM) [34], Deconvolutional network (DeCNN) [35] and convolutional auto-encoder (CAE) [36] are employed to compare with proposed pre-training strategy on the STL-10 [37] dataset when labeled data is limited. Specifically, we transform the data of STL-10 into 64×64 first. Then, the 100000 unlabeled images with these pre-training methods are used to pre-train our devised network as described in Table II. Finally, a small amount of images random sampled from STL-10 are utilized to fine-tune the CNN. As shown in

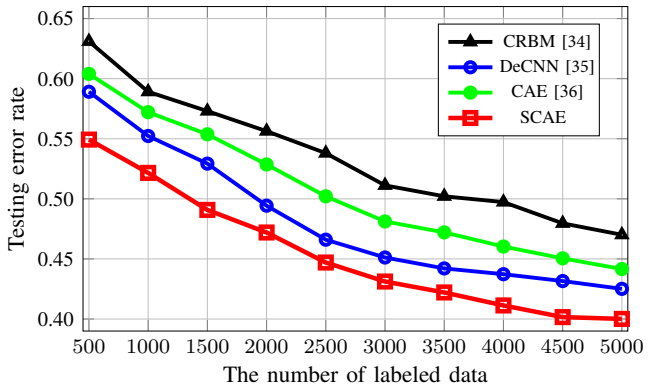


Fig. 7. The testing error rate of different methods with various number of labeled images on the STL-10 dataset.

Fig. 7, a good initial point is beneficial for training deep neural network when labeled data is limited. Actually, our proposed SCAE pre-training method can achieve best performance when labeled data is dramatically lacking.

6) Discussion

There are some advantages of the proposed model. Firstly, the developed pre-training method can initialize a group of good parameters to network by utilizing unlabeled data, this strategy can significantly improve the classification performance when labeled data is limited. Secondly, the proposed CNN model only requires less computation time and fewer resources than a deeper network. The advantage extends the practicability to practical applications.

VI. Conclusions

Acquiring reliable vehicle numbers and categories data is necessary to improve the quality of Intelligent Transport System. In this paper, we have proposed two effective and efficient models for vehicle detection and classification. For the vehicle detection, we focus on proposing a generalized approach to solve the problem of multiple-vehicle occlusion by analyzing the closure and convex hull of vehicles. Experimental results show that the proposed method can not only separate most of the vehicles independently when multiple-vehicle occlusion occurs, but meet the requirement of real-time processing. We also address the problem of vehicle classification. To this end, a CNN model has been presented for classification problem. Furthermore, based on the sparse coding and auto-encoder, a new pre-training strategy is presented to pre-train the model. This strategy makes the deep model work well when labeled data is limited. The experimental results demonstrate that our model achieves superior performance and can be implemented on the ordinary computer based on CPUs only.

Acknowledgment

The authors would like to thank the reviewers and the associate editor for their valuable suggestions that improved the manuscript significantly. This work is supported in part by the National Natural Science Foundation of China (NSFC Nos. 91646207, 61403376, 61370039 and 91338202), and the Beijing Nature Science Foundation under Grant No. 4162064.

References

[1] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[2] S. Gupte, O. Masoud, R. F. Martin, and N. P. Papanikolopoulos, "Detection and classification of vehicles," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 3, no. 1, pp. 37–47, 2002.

[3] C. C. R. Wang and J. J. J. Lien, "Automatic vehicle detection using local featuresła statistical approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 83–96, 2008.

[4] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 267–276, 2010.

[5] S. Cherng, C. Y. Fang, C. P. Chen, and S. W. Chen, "Critical motion detection of nearby moving vehicles in a vision-based driver-assistance system," *Intelligent Transportation Systems IEEE Transactions on*, vol. 10, no. 1, pp. 70–82, 2009.

[6] C. Wang, Y. Fang, H. Zhao, C. Guo, S. Mita, and H. Zha, "Probabilistic inference for occluded and multiview on-road vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 215–229, 2016.

[7] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 1773–1795, 2013.

[8] M. Liang, X. Huang, C. H. Chen, and X. Chen, "Counting and classification of highway vehicles by regression analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2878–2888, 2015.

[9] S. H. Yu, J. W. Hsieh, Y. S. Chen, and W. F. Hu, "An automatic traffic surveillance system for vehicle tracking and classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 175–187, 2006.

[10] T. Gandhi, R. Chang, and M. M. Trivedi, "Video and seismic sensor-based structural health monitoring: Framework, algorithms, and implementation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 169–180, 2007.

[11] B. T. Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 425–437, 2008.

[12] C. Ozkurt and F. Camci, "Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks," *Mathematical and Computational Applications*, vol. 14, no. 3, pp. 187–196, 2009.

[13] M. Kafai and B. Bhanu, "Dynamic bayesian networks for vehicle classification in video," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 100–109, 2012.

[14] Y. Peng, Y. Yan, W. Zhu, and J. Zhao, "Vehicle classification using sparse coding and spatial pyramid matching," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 259–263.

[15] X. Ma and W. E. L. Grimson, "Edge-based rich representation for vehicle classification," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1185–1192.

[16] C. C. C. Pang, W. W. L. Lam, and N. H. C. Yung, "A method for vehicle count in the presence of multiple-vehicle occlusions in traffic images," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 441–459, 2007.

[17] C. L. Huang and W. C. Liao, "A vision-based vehicle identification system," in *Pattern Recognition, International Conference on*, 2004, pp. 364–367.

[18] W. Zhang, Q. J. Wu, X. Yang, and X. Fang, "Multilevel framework to detect and handle vehicle occlusion," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 1, pp. 161–174, 2008.

[19] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *International IEEE Workshop on 3d Representation and Recognition*, 2013, pp. 554–561.

[20] L. Yang, P. Luo, C. Change Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3973–3981.

[21] Y. Zhou, H. Nejati, T.-T. Do, N.-M. Cheung, and L. Cheah, "Image-based vehicle analysis using deep neural network: A systematic study," *arXiv preprint arXiv:1601.01145*, 2016.

[22] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Citeseer, 1988, p. 50.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842*, 2014.

[24] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy *et al.*, "Deepid-net: Deformable deep convolutional neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2403–2412.

[25] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, p. 106, 1962.

[26] B. A. Olshausen *et al.*, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[29] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, 2013.

[30] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[31] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Networks for Machine Learning, 2012.

[32] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

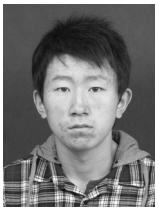
[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Science*, 2014.

[34] M. Norouzi, M. Ranjbar, and G. Mori, "Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2735–2742.

[35] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2528–2535.

[36] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 52–59.

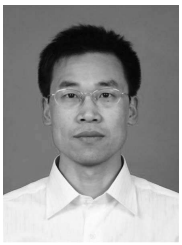
[37] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 215–223.



Jianlong Chang received the B.S. degree in School of Mathematical Sciences from University of Electronic Science and Technology of China, Chengdu, China, in 2015. Currently, he is a student in National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include deep learning, pattern recognition and machine learning.

Lingfeng Wang received his B.S. degree in computer science from Wuhan University, Wuhan, China, in 2007. He received his Ph.D. degree from Institute of Automation, Chinese Academy of Sciences in 2013. He is currently an associate professor with the National Laboratory of Pattern Recognition of Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision and image processing.

Gaofeng Meng received the B.S. degree in Applied Mathematics from Northwestern Polytechnical University in 2002, and the M.S. degree in Applied Mathematics from Tianjing University in 2005, and the Ph.D degree in Control Science and Engineering from Xi'an Jiaotong University in 2009. He is currently an associate professor of the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. He also serves as an Associate Editor of Neurocomputing. His current research interests include document image processing, computer vision and pattern recognition.



Shiming Xiang received the B.S. degree in mathematics from Chongqing Normal University, Chongqing, China, in 1993, the M.S. degree from Chongqing University, Chongqing, China, in 1996, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2004. From 1996 to 2001, he was a Lecturer with the Huazhong University of Science and Technology, Wuhan, China. He was a Postdoctorate Candidate with the Department of Automation, Tsinghua University, Beijing, China, until 2006. He is currently a Professor with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include pattern recognition, machine learning, and computer vision.

Chunhong Pan received his B.S. Degree in automatic control from Tsinghua University, Beijing, China, in 1987, his M.S. Degree from Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, China, in 1990, and his Ph.D. degree in pattern recognition and intelligent system from Institute of Automation, Chinese Academy of Sciences, Beijing, in 2000. He is currently a professor with the National Laboratory of Pattern Recognition of Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, image processing, computer graphics, and remote sensing.