International Journal of Control, Automation, and Systems (2015) 13(4):1020-1031 ISSN:1598-6446 eISSN:2005-4092
DOI 10.1007/s12555-014-0119-z http://www.springer.com/12555

# Domain Adaption of Vehicle Detector based on Convolutional Neural Networks

**Xudong Li, Mao Ye*, Min Fu, Pei Xu, and Tao Li**

**Abstract:** Generally the performance of a vehicle detector will decrease rapidly, when it is trained on a fixed training set but applied to a specific scene with view changes. The reason is that in the training set only a few samples are helpful for vehicle detection in the specific scene while other samples disturb the accurate detections. To solve this problem, we propose a novel transfer learning method to adapt the trained vehicle detector based on convolutional neural networks (ConvNets) to a specific scene with several new labeled samples. At first we reserve the share-filters and update the non-shared filters to improve the sensitivity of the vehicles in the specific scene. Then we combine the similar feature maps to accelerate the detection speed. At last for making the vehicle detector stable, we fine-tune it several times with the updated training set. Our contributions are an original research on transferring the vehicle detector based on ConvNets and an optimization approach about removing the redundant connections in the ConvNet vehicle detector. The extensive comparative experiments on three different datasets demonstrate that the transferred detectors achieve the improvements on both of the accuracy and speed.

**Keywords:** Convolutional neural networks, domain adaption, transfer learning, vehicle detection.

## 1. INTRODUCTION

Vehicle detection is an essential part of modern intelligent transportation monitoring system [1,2]. Many vehicle detection methods have been proposed in recent years. Roughly, most of them belong to two categories, i.e., based on features [3-5] and based on models [6-8].

One category focuses on extracting the exclusive features. Han *et al*. [3] take into account the spatial property of the image patch of vehicles and propose the extend HOG features which incorporate the spatial locality in the standard HOG features [9]. Cheng *et al*. [4] propose the boosted Gabor features whose parameters are learned from examples and optimized for each sub-window to get the good response for a vehicle part. Zheng *et al*. [5] consider relatively consistent structural components of vehicles and propose the image strip features which represent various kinds of basic local elements of vehicles, such as bumper, pillars, wheels, etc.

The other concentrates on designing the classifier. Jin *et al*. [6] use the morphological shared-weight neural network which incorporates both spatial and spectral characteristics to classify pixels into vehicles and non-vehicles. Wu *et al*. [7] propose the cluster boosted tree (CBT) which is automatically constructed for multi-view vehicle detection. This method employs the unsupervised clustering to divide the sample space, unlike the similar classifier model proposed by Huang *et al*. [10] which needs the predefined knowledge of the intra-class sub-categorization. Cheng *et al*. [8] use the dynamic Bayesian network (DBN) for the classification purpose. According to the extracted features, the trained DBN estimates the unknown state which predicts whether a pixel belongs to a vehicle or not.

However, all above methods adopt the hand-crafted features which are usually the low-level features so that they are not enough to represent the vehicles. In addition the more complex classifiers they use, the more time they take for training and detection.

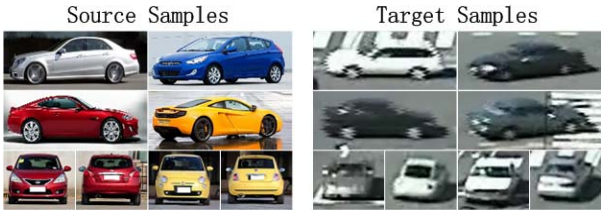
A desirable vehicle detector has the ability to learn features automatically and uses a simple classifier for efficiency. Fortunately, deep learning [11-13] can help us reach these goals. As an important branch of deep learning, convolutional neural networks (ConvNets) [14,15] have already been proved to make great successes. Garcia *et al*. [16] propose a face detection approach based on a convolutional neural architecture. Their key contribution is to show that a two-stage ConvNet obtains higher detection rate with a particularly low level of false positives than fully connected multi-layer perceptrons. For accelerating the detection speed, Chen *et al*. [17] design a face detector based on a ConvNet with a simple feature map and a coarse-to-fine classifier strategy. Sermanet *et al*. [18] use the

Manuscript received March 12, 2014; revised September 3, 2014; accepted September 29, 2014. Recommended by Associate Editor Dong-Joong Kang under the direction of Editor Euntai Kim.

This work was supported in part by the National Natural Science Foundation of China (61375038) and the Academic Support Program for Excellent Doctors (YBXSZC20131064).

Xudong Li, Mao Ye, Min Fu, Pei Xu, and Tao Li are with the School of Computer Science and Engineering, Center for Robotics and Key Laboratory for Neuro Information of Ministry of Education, University of Electronic Science and Technology of China, Chengdu 611731, P. R. China (e-mails: {lixudong268, cvlab.uestc, fumin268, xupei268, cvlablitao}@gmail.com).

* Corresponding author.




(a) Horizontal view. (b) Overhead view.

Fig. 1. Vehicle samples which are respectively selected from the source domain and the target domain (MIT traffic dataset).

unsupervised method based on convolutional sparse coding to pre-train the filters of ConvNet and achieve the state-of-the-art results of all major pedestrian datasets. Similarly, we can also obtain a good vehicle detector based on a single ConvNet, named the ConvNet vehicle detector.

When these vehicle detectors [3-8] and the ConvNet vehicle detector are applied to a specific scene, the performance always decreases rapidly. The specific scene only contains the vehicles with a few views, while in the training set many vehicles with other views are useless for the specific scene and even disturb the accurate detections. For easy of description, above vehicle detectors are called the source detectors which are trained by the source training samples. The target samples are captured from the specific scene which is referred to the target domain. The differences between the source samples and the target samples are shown in Fig. 1.

In this paper, we propose a novel transfer method to adapt the ConvNet vehicle detector to the target domain with only several labeled target samples. We use three steps to transfer the ConvNet vehicle detector, which are illustrated in Fig. 2. At first the shared filters are found by comparing the feature maps between the source domain and the target domain, while the non-shared filters are updated by the stochastic gradient descent algorithm. After this feature-level transfer, the detector becomes more sensitive to the vehicles in the target domain. Then we combine the similar feature maps in the detector to remove the redundant connections. In this way the detection speed is accelerated. Finally the fine-tuning step retrains the detector several times with the updated training set so that it becomes stable by learning the background information in the target domain and correcting the errors produced in the previous step.

Our contributions are : 1) A novel transfer method for the ConvNet vehicle detector is proposed. Though there exist some similar works, such as transferring the boosting-style detector [19], transferring the SVM-based pedestrian detector [20], etc., to the best of our knowledge, our work first transfers the ConvNet vehicle detector. 2) We propose an approximate optimization method which combines the similar feature maps in the detector for improving the detection speed.

The rest of this paper is arranged as follows. In Section 2, we briefly introduce the ConvNet vehicle detector and the procedure of vehicle detection. Section 3

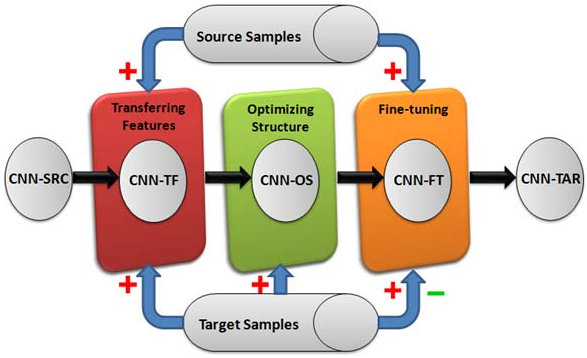



Fig. 2. The pipeline of our transfer learning method. The input CNN-SRC is a source vehicle detector. CNN-TF, CNN-OS and CNN-FT are the outputs in the three steps respectively. The final output CNN-TAR is the target vehicle detector. + and - represent the positive and negative samples respectively.

elaborates three steps of transferring the ConvNet vehicle detector. Extensive experiments and discussions on three datasets are given in Section 4. In Section 5, we summarize our paper and discuss the future work.

## 2. THE CONVNET VEHICLE DETECTOR

The ConvNet vehicle detector is a hierarchical model that can learn features automatically in a supervised way. The input is an image, and the output is a value produced by only one neural node standing for the class label, for instant 1 for a vehicle and 0 for the background. The ConvNet vehicle detector consists of two parts. The first part is a multi-stage feature extractor. The second part is a classifier which is a full connected neural network without hidden layers.

The ConvNet vehicle detector can learn good representations because its first part extracts features from the low level to the high level stage by stage [21]. Without loss of generality, we state the computations in the $k$th stage where $k = 1,2,3$. Suppose the set of input feature maps is $x^k = \{x_m^k \mid m = 1,\cdots,R^{k-1}\}$, and the set of output feature maps is $z^k = \{z_n^k \mid n = 1,\cdots,R^k\}$, where $R^{k-1}$ and $R^k$ are the cardinalities of the sets $x^k$ and $z^k$ respectively. Since each stage includes a convolutional layer and a subsampling layer, the set of middle feature maps between these two layers is denoted as $y^k = \{y_n^k \mid n = 1,\cdots,R^k\}$. Each feature map $y_n^k$ is computed in the convolutional layer as

$$y_n^k = F\left(\sum_{m=1}^{R^{k-1}} x_m^k \otimes f_{nm}^k + b_n^k\right), \tag{1}$$

where $\otimes$ stands for the convolution operation of the feature map $x_m^k$ with the filter $f_{nm}^k$ and $b_n^k$ represents a bias. The non-saturating nonlinearity function $F(x) = \max(0,x)$ is elected as the activation function. Following the convolutional layer, the subsampling layer uses a mean kernel of fixed size scanning each feature

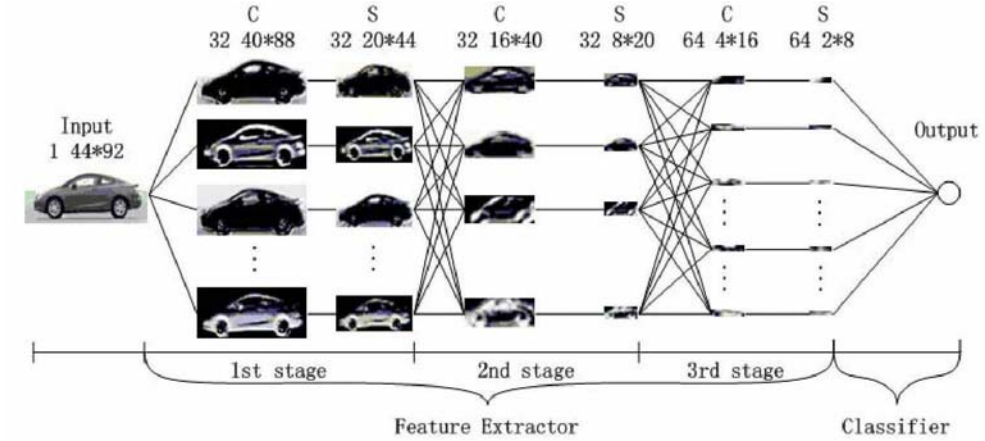



Fig. 3. The architecture of the ConvNet vehicle detector for detecting the vehicles with the profile view. This detector has a three-stage feature extractor. C stands for the convolutional layer and S represents the subsampling layer.

map $y_n^k$ without overlapping to form the corresponding feature map $z_n^k$ as the follows:

$$z_n^k = \text{subsampling}(y_n^k). \tag{2}$$

The corresponding filter set respect to the feature map $z_n^k$ is $f_n^k = \{f_{nm}^k \mid m = 1, \cdots R^{k-1}\}$. In our paper when $k = 1$, $R^{k-1} = 1$. Then the output feature map set $z_n^k$ is used as the input of the next stage. For the input image $x^0$, $z_n^k(x^0)$ is denoted as the $n$ th output feature map in the $k$ th stage.

The procedure of our vehicle detection method is a little different with the traditional detection methods [22,23,36]. Firstly the entire image is entered into the first part of a trained ConvNet vehicle detector to extract its high-level features. Secondly in order to improve the detection speed, we slide a fixed-size window on the feature maps in the last stage with overlapping simultaneously instead of the input image. Thirdly the features in the scanned window are send to the second part to predict whether there exists a vehicle. At last the non-maximum suppression method is employed to combine the overlapped bounding boxes.

However, we have not yet found a practical way to detect multi-view vehicles by one ConvNet due to their different input sizes. Therefore we design two detectors, one for detecting the vehicles with the profile view and another one for detecting the vehicles with the frontal and rear views. Fig. 3 shows the architecture of the ConvNet vehicle detector for detecting the profile-viewed vehicles.

## 3. TRANSFERRING THE CONVNET VEHICLE DETECTOR

Let $\mathscr{S}^+ = \{s_i^+\}_{i=1}^S$ be the positive samples in the source domain and $\mathscr{T}^+ = \{t_j^+\}_{j=1}^T$ be the positive samples in the target domain, where $S$ is large and $T$ is small. For obtaining a good detector in the target domain, we need to adjust the source detector to the target domain by using three steps: transferring features, optimizing structure and fine-tuning.

### 3.1. Transferring features

The source detector has stored the appearance knowledge of the vehicles in the form of the filters. Some filters belong to the shared filters which represent the common structure of the vehicles across domains. The other filters are the non-shared filters which do not match the target domain. Therefore the first step of our transfer learning is the feature-level transfer which reserves the shared filters and updates the non-shared filters. Considering that the low-level features are the basis of the high-level features, we first transfer the low-level features, then the high-level ones.

To search the shared filters, we need to compare the feature maps from different domains. Assume that the filter set $f_n^k$ contains the shared filters, then the feature maps $z_n^k(\mathscr{S}^+)$ should be similar with the feature maps $z_n^k(\mathscr{T}^+)$. However, since the vehicles in $\mathscr{S}^+$ have more views than the vehicles in $\mathscr{T}^+$, a part of the feature maps $z_n^k(\mathscr{S}^+)$ which cannot possibly appear in the target domain are useless for the feature-level transfer. Hence $\mathscr{S}^+$ should be divided into $C$ classes by K-mean clustering algorithm, $\mathscr{S}^+ = \bigcup_{c=1}^C \mathscr{S}_c^+$, so that each class $\mathscr{S}_c^+$ is the set of the vehicles with the same view from which the similar feature maps $z_n^k(\mathscr{S}_c^+)$ can be extracted by the source detector. Then the average feature map of each class is computed as

$$\overline{z}_n^k(\mathscr{S}_c^+) = \frac{1}{\left|\mathscr{S}_c^+\right|} \sum_{s_c^+ \in \mathscr{S}_c^+} z_n^k(s_c^+), \tag{3}$$

where $|\bullet|$ represents the set cardinality. In order to find a corresponding feature map in $C$ classes for a target image $t_j^+, j = 1, \cdots, T$, we find the class $\hat{c}$ which has the maximum average similarity with $t_j^+$

$$\hat{c}(j) = \arg\max_c \frac{1}{\left|\mathscr{S}_c^+\right|} \sum_{s_c^+ \in \mathscr{S}_c^+} \cos(t_j^+, s_c^+), \tag{4}$$

where $\cos(\bullet,\bullet)$ represents the cosine similarity between two images. After that, we estimate whether $f_n^k$ is the shared filter set for each $t_j^+$ by the following rule,

$$\text{if } \cos(z_n^k(t_j^+), \bar{z}_n^k(\mathscr{S}_{\hat{c}(j)}^+)) \begin{cases} \geq \varepsilon \\ < \varepsilon, \end{cases}$$
$$\text{then } f_n^k \text{ is} \begin{cases} \text{shared} \\ \text{non-shared}, \end{cases} \tag{5}$$

where $\varepsilon$ is a decision threshold. If most of the samples in $\mathscr{T}^+$ agree that $f_n^k$ is the shared filter set, the filters in $f_n^k$ will be reserved; otherwise the filters in $f_n^k$ need to be updated.

The way to updated the non-shared filter set $f_n^k$ is to make the feature map $z_n^k(t_j^+)$ close to its corresponding feature map $\bar{z}_n^k(\mathscr{S}_{\hat{c}(j)}^+)$ so that the parameter of the classifier can be used in the target domain. Therefore we define the loss function

$$L = \frac{1}{T}\sum_{j=1}^{T}\left\| z_n^k(t_j^+) - \bar{z}_n^k(\mathscr{S}_{\hat{c}(j)}^+)\right\|_2^2 \tag{6}$$

and employ the stochastic gradient descent algorithm to update the non-shared filter set $f_n^k$.

### 3.2. Optimizing structure

When the vehicle detector is based on a large ConvNet, there exist some similar feature maps in each stage which can be considered as the redundant structure. Especially after transferring features, the redundant structure is formed with high probability. Hence for accelerating the detection speed, the second step is to optimize the structure by combining the similar feature maps. Since the redundant structure is always easily formed in the last stage where there are many small-size feature maps, the order of optimizing structure will be from the last stage to the first stage.

In order to find the similar feature maps in the *k*th stage, we compute a similarity matrix $\phi_j^k$ whose elements represent the cosine similarities between the feature maps of the sample $\mathrm{t}_j^+, j = 1, \cdots, T$,

$$\phi_j^k(p,q) = \cos(z_p^k(t_j^+), z_q^k(t_j^+)), \tag{7}$$

where $p$ and $q$ are the indexes of the corresponding feature maps. Then the average similarity matrix of the target domain is calculated as

$$\bar{\phi}^k = \frac{1}{T}\sum_{j=1}^{T}\phi_j^k. \tag{8}$$

According to $\bar{\phi}^k$, the feature maps are clustered by employing the method of analytic hierarchy process (AHP) [24]. The feature maps in the same cluster should be combined so that each class retains only one feature map.

Suppose the feature map $z_p^k$ and the feature map $z_q^k$ belong to the same cluster, and $z_q^k$ is absorbed by $z_p^k$. Since the filter set $f_q^k$ is only used for forming $z_q^k$ according to (1) and (2), $f_q^k$ can be deleted when $z_q^k$ is replaced by $z_p^k$. On the other hand, the filters which connect with $z_q^k$ in the next stage cannot be deleted directly. If $y_r^{k+1}$ is the *r*th feature map in the $(k+1)$ th stage, then according to (1)

$$y_r^{k+1} = F(\cdots + z_p^k \otimes f_{rp}^{k+1} + z_q^k \otimes f_{rq}^{k+1} + \cdots + b_r^{k+1}). \tag{9}$$

As we let $z_p^k \approx z_q^k$, then

$$y_r^{k+1} \approx F(\cdots + z_p^k \otimes (f_{rp}^{k+1} + f_{rq}^{k+1}) + \cdots + b_r^{k+1}). \tag{10}$$

If let $f_{rp}^{k+1} \leftarrow f_{rp}^{k+1} + f_{rq}^{k+1}$, equation (10) can be reformulated as

$$y_r^{k+1} \approx F(\cdots + z_p^k \otimes f_{rp}^{k+1} + \cdots + b_r^{k+1}). \tag{11}$$

Therefore the filters connecting to $z_q^k$ in the next stage can be deleted after adding them to a new filter. Note that if we combine the feature maps in the last stage, the convolution $\otimes$ should be replaced by the matrix dot product, and $f_{rp}^{k+1}$ and $f_{rq}^{k+1}$ are a part of weights of the classifier.

Although optimizing structure can accelerate the detection speed, it inevitably will produce a few errors due to the approximate combination in (10).

### 3.3. Fine-tuning

After above two steps, the optimized vehicle detector still cannot perform well in the target domain on the account of the fact that it has not yet learned the background knowledge (negative samples) of the target domain and is interfered by a few errors produced by the approximate structure optimization. Hence the third step is to fine-tune the whole detector with the updated training set.

Due to lacking of enough $\mathscr{T}^+$, the detector cannot be fine-tuned well. Fortunately some vehicles in $\mathscr{S}^+$, which have the similar distribution with the vehicles in $\mathscr{T}^+$, can extend the training set. Pang *et al*. [19] use the ratio of the target density to the source density to be the availability $\lambda_i^+$ for $s_i^+, i = 1, \cdots, S$. Furthermore $\lambda_i^+$ can be reformulated with the conditional probability

$$\lambda_i^+ = \frac{p(s_i^+ \mid \mathscr{T}^+)}{p(s_i^+ \mid \mathscr{S}^+)} = \frac{p(\mathscr{T}^+ \mid s_i^+)p(\mathscr{S}^+)}{p(\mathscr{S}^+ \mid s_i^+)p(\mathscr{T}^+)}, \tag{12}$$

where $p(s_i^+ \mid \mathscr{S}^+)$ and $p(s_i^+ \mid \mathscr{T}^+)$ represent the probability of $s_i^+$ in the source domain and in the target domain respectively. In our paper these conditional probabilities are modeled as follows,

$$p(\mathscr{T}^+ \mid s_i^+) = \frac{1}{1 + (o_t(s_i^+) - 1)^2},$$
$$p(\mathscr{S}^+ \mid s_i^+) = \frac{1}{1 + (o_s(s_i^+) - 1)^2}, \tag{13}$$

where $o_t(s_i^+)$ and $o_s(s_i^+)$ represent the output value of $s_i^+$ from the previous transferred detector and the source detector respectively. If $s_i^+$ can be classified correctly across domains, in (13) these two conditional probabilities will be close to 1. $p(\mathscr{S}^+)$ and $p(\mathscr{T}^+)$ are assumed to be equal because we have the same probability to observe the vehicles with different views. Therefore $\lambda_i^+$ is reformulate as

$$\lambda_i^+ = \frac{1+(o_s(s_i^+)-1)^2}{1+(o_t(s_i^+)-1)^2}. \quad (14)$$

If $\lambda_i^+ \geq 1$, $s_i^+$ has the similar view with the vehicles in the target domain and can help fine-tune the detector. The samples in $\mathscr{S}^+$ with $\lambda_i^+ \geq 1$ are along with $\mathscr{T}^+$ to constitute the new positive sample set $\mathscr{D}^+$. Although $\mathscr{D}^+$ is enriched, the number of $\mathscr{D}^+$ is less than the number of negative samples $\mathscr{T}^-$ in the target domain. In order to obtain a balanced training set, we only pick hard negative samples from the target domain, which own high output values, to constitute the new negative sample set $\mathscr{D}^-$. The training set $\mathscr{D}$ is obtained by combining $\mathscr{D}^+$ and $\mathscr{D}^-$. In each iteration we compute the availability of each vehicle in $\mathscr{S}^+$ and pick new hard negative samples to update the training set $\mathscr{D}$. There-fore the optimized detector is obtained after being retrained with the updated training set $\mathscr{D}$ until convergence. The whole algorithm is described in Algorithm 1.

**Algorithm 1:** The whole algorithm of transferring the ConvNet vehicle detector.

1. **Input:** the source ConvNet vehicle detector CNN-SRC, the positive samples in the source domain $\mathscr{S}^+$, the positive samples in the target domain $\mathscr{T}^+$ and the negative samples in the target domain $\mathscr{T}^-$.
2. **Output:** the target ConvNet vehicle detector CNN-TAR.
3. CNN-FT ← CNN-SRC.
4. Finding the corresponding feature maps for each sample in $\mathscr{T}^+$.
5. **for** $k = 1, 2, 3$ **do**
6. Searching the shared filters as in Sec.3.1.
7. **repeat**
8. Updating the non-shared filters of CNN-FT.
9. **until** Performing well on the validation set.
10. **end for**
11. CNN-OS ← CNN-TF.
12. **for** $k = 3, 2, 1$ **do**
13. Computing the average similarity map $\overline{\phi}^k$.
14. Finding the similar feature maps.
15. Combining the similar feature maps of CNN-OS as in Sec.3.2.
16. **end for**
17. CNN-TF ← CNN-OS.
18. **for** $iter = 1, \cdots$ **do**
19. Updating training set $\mathscr{D}$ as in Sec.3.3.
20. **repeat**
21. Retraining CNN-TF with $\mathscr{D}$.
22. **until** Convergence.
23. **end for**
24. CNN-TAR ← CNN-TF.

## 4. EXPERIMENTS

The experiments consist of three parts to illustrate the efficiency of our algorithm. Because as far as we know there does not exist a vehicle detector based on a ConvNet, Section 4.1 describes the details about two source ConvNet vehicle detectors and demonstrates their performances on Caltech-101 car dataset [25] and Caltech 1999 cars dataset [26]. In Section 4.2 the proposed algorithm is evaluated on three dataset: UIUC car dataset [27], MIT traffic dataset [28] and our road dataset. A series of analyses of our algorithm are shown in Section 4.3.

### 4.1. Source ConvNet vehicle detector

For training two source ConvNet vehicle detectors, we collect a plenty of vehicle samples which are roughly divided into two classes: the profile-view vehicles and the frontal&rear-view ones. The profile-view vehicles contain 1200 samples which are normalized into the size of 44×92 pixels, and the frontal&rear-view vehicles include 1200 samples which are resized to 44×44 pixels. Each vehicle is located in the center of the image and surrounded by several pixels as the background. Referring to Sermanet *et al*. [18], we add 3 transformed versions of each vehicle by translation ([-2,+2] pixels), scale ([0.9,1.1] ratio) and rotation ([-10,+10] degrees) randomly to expand the samples. Moreover the profile-view vehicles are flipped horizontally. Thus there are 9600 profile-view vehicles and 4800 frontal&rear-view ones in total.

As described in Section 2, the source detector CNN-SRC-P detects the profile-view vehicles and the source detector CNN-SRC-F&R detects the frontal&rear-view vehicles. Both of them adopt the same feature extractor described by Krizhevsky *et al*. [15] (layers-80sec.cfg). The architecture details of CNN-SRC-P are shown in Fig. 3. We employ the training-bootstrapping algorithm [16] to train these source detectors. The initial training set includes an equal number of positive and negative samples. 5 bootstrapping passes are preformed. In each pass 600 vehicles and 600 false positives captured from 1200 scenery images (containing no vehicle) which are collected from INRIA person dataset [9] and INRIA car dataset [29] are added to the current training set. All training procedure adopts mini-batch SGD on batches of 100 images with the learning rate fixed at 0.01.

To demonstrate the performance of source ConvNet vehicle detectors, CNN-SRC-P is tested on Caltech-101 car dataset [25] and CNN-SRC-F&R is evaluated on Caltech 1999 cars dataset [26]. Caltech-101 car dataset, which is the subset of Caltech-101 dataset, consists of 123 gray-scale images with the same size 197×300. Each image contains one profile-view vehicle of size 40×100. Caltech 1999 cars dataset contains 126 color images with the same size 592×896. Each image includes one rear-view vehicle which is approximately normalized into the size of 355×435 pixels.

We compare CNN-SRC-P with two methods [30,31] on Caltech-101 car dataset (see Fig. 4). Xu *et al*. [30] used voting spaces trained by few samples to detect objects. Their method proposed a new feature to represent local principal gradient orientation information and constructed voting spaces based on a few samples. Yang *et al*. [31] showed the contour-based method which

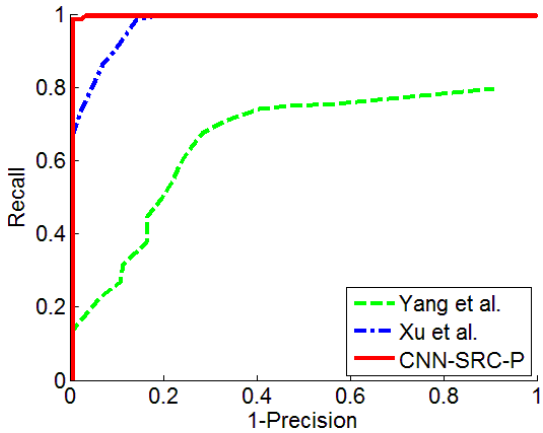



Fig. 4. Recall-Precision curves of different vehicle detectors on Caltech-101 car dataset.

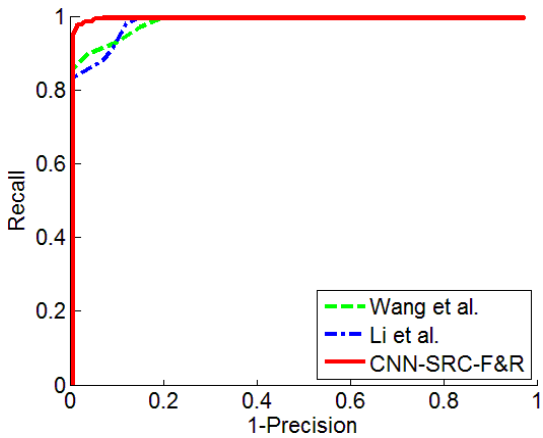
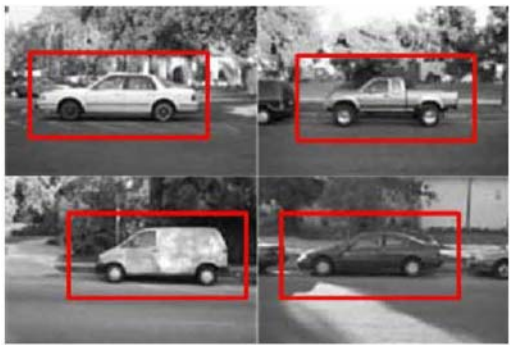



Fig. 5. Recall-Precision curves of different vehicle detectors on Caltech 1999 cars dataset.

considered object detection as a matching problem between model segment and image. Their method reduced missing segments by finding dominant sets in weighted graphs. However, the accuracy of these works [30,31] seriously depends on the reliability of samples.

CNN-SRC-F&R is compared with other two methods [32,33] on Caltech 1999 cars dataset (see Fig. 5). Li *et al*. [32] presented a Hough context model for detecting a certain object class. Their method improved the detection accuracy by combining context information into Hough spaces. Wang *et al*. [33] automatically detected vehicles through a statistical approach. In order to reduce the effect of geometric variance and partial occlusion, their method used local features from several significant subregions of image. However, the complexity of these models [32,33] is relatively high due to the hand-crafted features.

Figs. 4 and 5 show that our source ConvNet vehicle detectors outperform the previous methods [30-33] obviously. The results demonstrate that source ConvNet vehicle detectors successfully learn the features of vehicles and can achieve high accuracy with low false positive rate in the case of using a simple classifier. Some detection results of source ConvNet vehicle detectors are displayed in Fig. 6. The outputs of red



(a) The results of CNN-SRC-P on Caltech-101 car dataset.

(b) The results of CNN-SRC-F&R on Caltech 1999 cars dataset.

Fig. 6. A part of detection results of source ConvNet vehicle detectors.

bounding boxes in Fig. 6(a) and Fig. 6(b) are more than 0.9 and 0.8 respectively.

However, when source ConvNet vehicle detectors are tested on other datasets, their performance is not as well as them on above two datasets due to some view changes or distortions (see Fig. 8, Fig. 10 and Fig. 12). Thus we apply our algorithm for transferring these two source ConvNet vehicle detectors to different target domains in the following experiments.

### 4.2. Target ConvNet vehicle detector

In this part we show the performance of our algorithm on three datasets. The Recall-Precision curve and the receiver operating characteristic (ROC) curve are used as the evaluation metric. A detection is regarded as right, if and only if the bounding box overlaps with an annotation more than 75% on UIUC car dataset or 50% on MIT traffic dataset and our road dataset. 10-fold cross-validation is used to find the proper parameters and validate the performance of vehicle detectors.

#### 4.2.1 UIUC car dataset

The first target domain is UIUC car dataset [27] which is usually used for evaluating the performance of the profile-view vehicle detectors. Its training set contains 550 positive samples and 500 negative samples. Its test set includes 170 gray-scale images containing 200 single-scale vehicles with the resolution of 100×40 pixels and 108 gray-scale images containing 139 multi-scale vehicles. Thus the positive samples of target

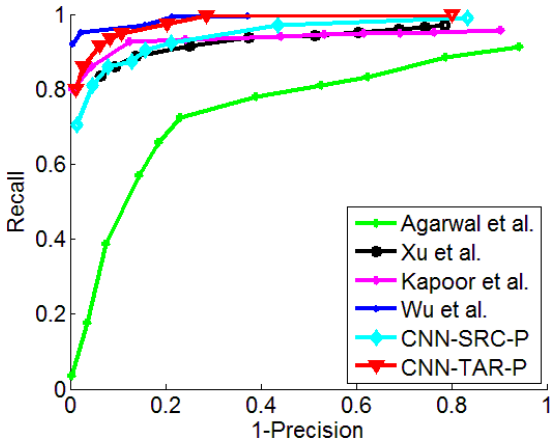
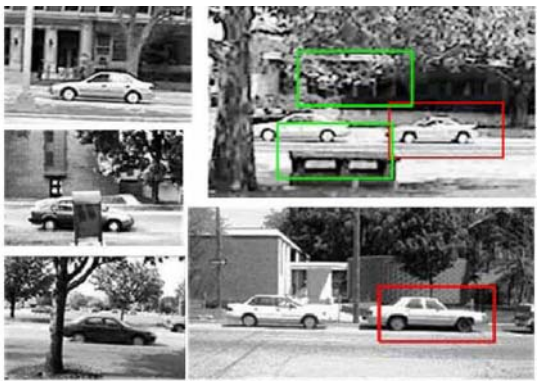



Fig. 7. Comparison of Recall-Precision curves between our method and other methods [27,30,34,35] on the single-scale UIUC car dataset. CNN-TAR-P denotes our target ConvNet vehicle detector for detecting the profile-view vehicles.

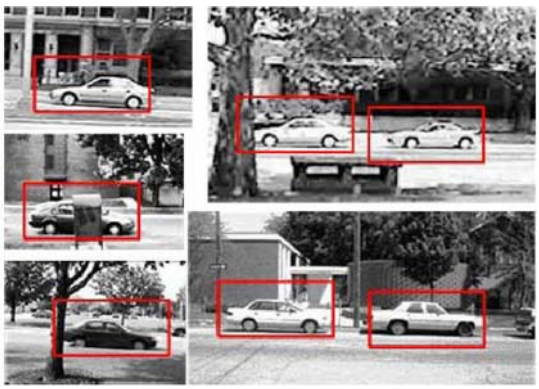
domain are 550 vehicles from UIUC training set. The negative samples of target domain not only include 500 negative samples in UIUC training set, but also contain a large number of negative samples which are used for training source vehicle detectors, because the background of target domain is not fixed.

To prove the performance improvements of our algorithm, we make a comparison between our method and some state-of-the-art methods [27,30,34,35] on the single-scale UIUC car dataset (see Fig. 7). Agarwal *et al*. [27] developed a learning-based approach which used a sparse, part-based representation and an automatically learning method to detect objects of interest. Kapoor *et al*. [34] introduced a conditional model for simultaneous part-based detection which learned a set of parts from a training set of images with segmentation masks. Wu *et al*. [35] showed a method, which adopted edgelet features for designing base detectors and employed boosting algorithm to learn the ensemble classifier with a cascade decision strategy, to simultaneously detect and segment objects. These methods [27,34,35] are all based on the training approaches which our method also belongs to, therefore it is meaningful to compare our method with them.

By observing the Recall-Precision curves in Fig. 7, the performance of source detector CNN-SRC-P is at the middle level. Its performance is better than two methods [27,30], but its recall is lower than the other two methods [34,35] in the case of the high precision. After transferring the source detector CNN-SRC-P, the target detector CNN-TAR-P is superior to the source detector CNN-SRC-P. Furthermore the performance of CNN-TAR-P surpasses that of the method [34] and gets very close to the best performance [35]. The performance improvements illustrate that our target detector has adapted the target domain. Fig. 8 shows the comparisons of some representative detection results produced by source detector and target detector on single-scale UIUC car dataset.



(a) The detection results of the source detector.



(b) The detection results of the target detector.

Fig. 8. Comparisons of some representative detection results on the single-scale UIUC car dataset. The red bounding box represents true positive and the blue one denotes false positive.

Table 1. Detection equal-error rates (EER) on the multi-scale UIUC car dataset.

| Method | EER |
|---|---|
| Agarwal *et al*. [27] | 44.08% |
| Seo *et al*. [23] | 77.66% |
| Xu *et al*. [30] | 91.34% |
| Kappor *et al*. [34] | 93.50% |
| CNN-SRC-P | 91.62% |
| CNN-TAR-P | **94.75**% |

Table 1 shows the comparisons of detection equal-error rates (EER) [23] on the multi-scale UIUC car dataset. Here, we also compare our method with several training-based methods [23,27,30,34] which have been described before. From Table 1, it can be observed that EER of CNN-TAR-P is higher than that of CNN-SRC-P about 3%. In addition, compared with these training-based methods, CNN-TAR-P achieves the best performance of EER. It demonstrates the efficiency of our method.

4.2.2 MIT traffic dataset

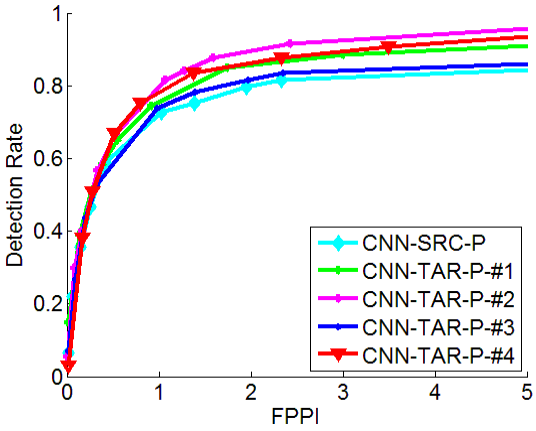
The second target domain is MIT traffic dataset [28] which includes a traffic video. The video is captured in a cross road by a fixed camera at the resolution of 720×480 pixels and is cut into 20 clips. The positive samples, which are extracted from the first five clips, include 400 profile-view vehicles and 355 frontal&rear-

Table 2. The target ConvNet vehicle detectors transferred by different steps. #1, #2 and #3 stand for the target detectors which are lack of a certain step during transfer learning procedure respectively. #4 denotes the target detector which is transferred by whole transfer learning procedure.

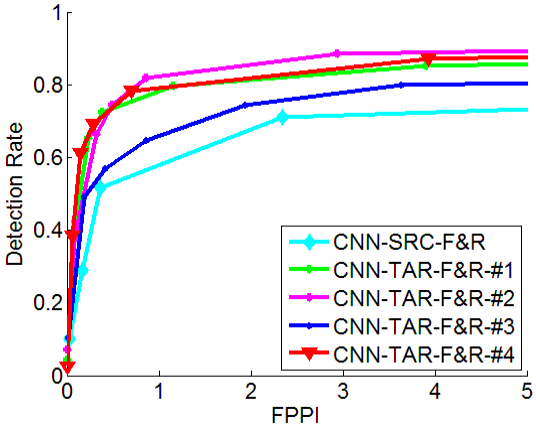
| Target detector | Source detector | Transferring features | Optimizing structure | Fine-tuning |
|---|---|---|---|---|
| CNN-TAR-P-#1 | CNN-SRC-P | × | √ | √ |
| CNN-TAR-P-#2 | | √ | × | √ |
| CNN-TAR-P-#3 | | √ | √ | × |
| CNN-TAR-P-#4 | | √ | √ | √ |
| CNN-TAR-F&R-#1 | CNN-SRC-F&R | × | √ | √ |
| CNN-TAR-F&R-#2 | | √ | × | √ |
| CNN-TAR-F&R-#3 | | √ | √ | × |
| CNN-TAR-F&R-#4 | | √ | √ | √ |

view vehicles. The negative samples are extracted at random from a few frames which do not contain vehicles. Our algorithm is tested on the residual clips from mv2_006.avi to mv2_020.avi. Unfortunately MIT traffic dataset only has the annotations of pedestrians, therefore we label the vehicles every 100 frames. Finally there are 1230 frames with 1586 profile-view vehicles and 1542 frontal&rear-view ones in total.

In order to verify the effects of each step of our algorithm, we transfer four different target detectors based on the same source detector for each view (see Table 2). The ROC curves of these vehicle detectors are plotted in Fig. 9. As expected, the performance of target detectors is better than that of source detectors. However, the best detectors are #2 instead of #4 according to Fig. 9. The reason is that it is inevitable to lose some vehicle information in the second step of our algorithm so that the fine-tuning step cannot enough correct the errors perfectly. Nevertheless the detection speed of #4 detectors is faster than that of #2 detectors (see in Section 4.3). Therefore, #4 detectors are considered as the best practical detectors based on the comprehensive comparisons. The accuracy of #1 detectors is improved, which illustrates the effects of fine-tuning. Moreover the gap of the accuracy between #4 detectors and #1 detectors demonstrates the important role of transferring features. However, #3 detectors are worst in all target detectors. Although the feature-level transfer makes them be sensitive to the vehicles of target domain, the lack of background knowledge of target domain leads to lots of false positives. The above comparisons prove that our method can transfer the ConvNet vehicle detector successfully with improvements on the accuracy.




(a) The ROC curves of detectors which detect the profile-view vehicles.




(b) The ROC curves of detectors which detect the frontal&rear-view vehicles.

Fig. 9. The ROC curves of different detectors on MIT traffic dataset.

Fig. 10 shows the comparisons of some detection results on MIT traffic dataset. Fig. 10(a) and Fig. 10(c) are the results of source detectors. A half of vehicles are missed, and there exist some false positives. Fig. 10(b) and Fig. 10(d) are the results of #4 target detectors. Obviously the number of false positives are reduced greatly, and most of vehicles are found, furthermore target detectors locate vehicles more precisely.

4.2.3 Our road dataset

The third target domain is our road dataset which contains 400 pictures filmed by a fixed camera in a very busy road. We utilize the first 50 pictures which include 300 frontal&rear-view vehicles as the training set for transfer learning and test our algorithm on the rest of 350 pictures.

In Fig. 11 we compare our algorithm with two methods [30,32] which have been compared with our source detectors in Section 4.1. Fig. 11 shows that source detector is a little better than one method [30] but is worse than the other method [32]. The reason is that the detectors of these two methods are reconstructed based on the target training set. However, after transferring source detector, target detector achieves the best performance. Especially when FPPI is low, the detection

Fig. 10. A part of detection results on MIT traffic dataset. The results of CNN-SRC-P and CNN-TAR-P-#4 are shown in (a) and (b) respectively. The results of CNN-SRC-F&R and CNN-TAR-F&R-#4 are shown in (c) and (d) respectively. The outputs of these bounding boxes are all over 0.5. The comparisons of the first column illustrate the reduction of false positives. The comparisons of the second column show the precise locations of target detectors. The comparisons of third and fourth columns demonstrate the improvements on the accuracy.

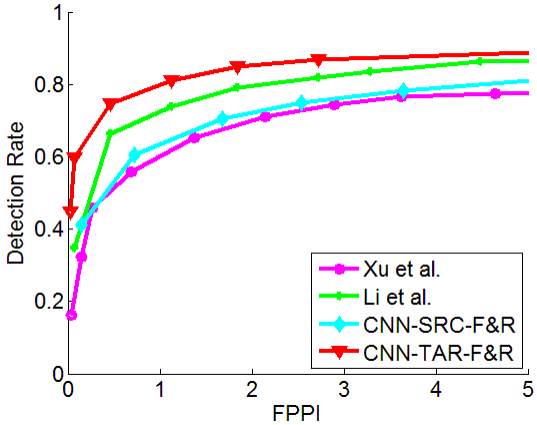



Fig. 11. Comparisons of the ROC curves on our road dataset.

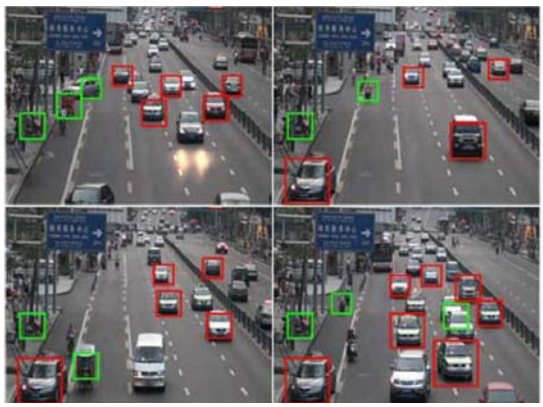
rate of the target detector is the double of that of contrastive methods. Fig. 12 shows the comparisons of some detection results between the source detector and the target detector.

4.3. The analysis of algorithm

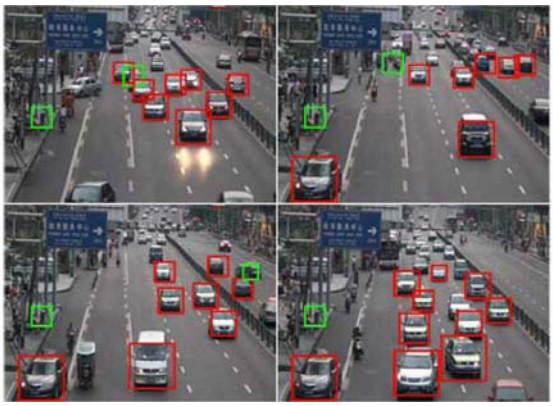
Now, the contrast experiments between the source detectors and the target detectors are presented to analyze our algorithm during transfer learning.

In order to analyze the degree of the feature-level transfer, Table 3 shows the percentage of the shared filters in each stage. As is apparently revealed in Table 3, the shared filters always appear in the 2nd and 3rd stages and their percentage in the 3rd stages is higher than that in the 2nd stage. The reason is that the low-level features are sensitive to variations caused by view changes and distortions, while the high-level features have the abilities to handle these variations in some degree. Thus all filters in the 1st stage are updated and the most of filters in the 2nd and 3rd stages are remained in the first step.

For analyzing the complexity of the ConvNet vehicle detector, the number of the feature maps in each stage is taken as the measure of the complexity. The changes of the complexity in each stage are reported in Table 4. Obviously the complexity of target detectors is lower than that of source detectors. According to our results, only several feature maps are combined in the 1st stage. It illustrates that we extract various kinds of low-level features for vehicle detection. In the 3rd stage about one-sixth of feature maps are combined. It demonstrates that



(a)



(b)

Fig. 12. Comparisons of the detection results on our road dataset in the condition that FPPI is 2.

Table 3. The percentage of the shared filters in each stage. Each row represents a target detector in a certain target domain.

| Target domain | View point | 1ST | 2ST | 3ST |
|---|---|---|---|---|
| UIUC | Profile | 0% | 87.5% | 92.2% |
| MIT | Profile | 0% | 62.5% | 87.5% |
| MIT | frontal&rear | 0% | 43.8% | 90.6% |
| road | frontal&rear | 0% | 53.1% | 85.9% |

Table 4. The changes of the complexity in each stage. The number in the bracket stands for the number of combined feature maps.

| Domain | View point | 1ST | 2ST | 3ST |
|---|---|---|---|---|
| Source | Profile | 32 | 32 | 64 |
| Target(UIUC) | Profile | 30(2) | 25(7) | 53(11) |
| Target(MIT) | Profile | 31(1) | 28(4) | 52(12) |
| Source | frontal&rear | 32 | 32 | 64 |
| Target(MIT) | frontal&rear | 29(3) | 24(8) | 51(13) |
| Target(road) | frontal&rear | 28(4) | 26(6) | 50(14) |

Table 5. Comparisons of the detection time between the source detectors and the target detectors.

| Domain | View point | Detection time |
|---|---|---|
| Source | Profile | 43.40±0.16ms |
| Target(UIUC) | Profile | 31.12±0.26ms |
| Target(MIT) | Profile | 33.19±0.23ms |
| Source | frontal&rear | 17.15±0.14ms |
| Target(MIT) | frontal&rear | 12.35±0.17ms |
| Target(road) | frontal&rear | 11.98±0.15ms |

the redundant structure will be formed with high probability when the size of the feature maps is small.

The detection speed is inversely proportional to the complexity of the detector. For the sake of contrastive analysis, the detection time of a single sample is used for measuring the promotion of the detection speed. Our experiment platform is Matlab. Table 5 shows the comparisons of the detection time between source detectors and target detectors. For testing the speed, we randomly select the equal number of positive samples and negative samples from the target domain and compute the average time for each sample. After optimizing structure, the detection speeds of the ConvNet vehicle detectors promote at least 23.5%.

## 5. CONCLUSION

In this paper, we apply the ConvNets to the task of vehicle detection and introduce a novel algorithm to transfer a ConvNet vehicle detector to target domain. Our transfer method contains three steps. Firstly a feature-level transfer, which finds the shared filters and updates the non-shared filters, is used to make the source detector more sensitive to the vehicles in target domain. Then in order to accelerate the detection speed, we remove the redundant structures by combining the similar feature maps. Last, we fine-tune the optimized detector with the updated training set several times so that the target detector learns the background of target domain and corrects the errors produced by the second step. Experiment results on three datasets confirm the good performance of our approach. However, our algorithm also exist two defects. The first one is that the target samples should be labeled beforehand. In some cases, the labeled samples in target domain may be unavailable. The second one is the approximate optimizing approach which produces some errors that influence the performance of vehicle detectors. Our future works are to find the appropriate way to solve these two defects.

## REFERENCES

[1] T. H. Chen, Y. F. Lin, and T. Y. Chen, "Intelligent vehicle counting method based on blob analysis in traffic surveillance," *Proc. of IEEE Innovative Computing, Information and Control*, pp. 238-238, 2007.

[2] X. Pan, Y. Guo, and A. Men, "Traffic surveillance system for vehicle flow detection," *Proc. of IEEE Computer Modeling and Simulation*, pp. 314-318, 2010.

[3] F. Han, Y. Shan, R. Cekander, H. S. Sawhney, and R. Kumar, "A two-stage approach to people and vehicle detection with hog-based SVM," *Proc. of Performance Metrics for Intelligent Systems 2006 Workshop*, pp. 133-140, 2006.

[4] H. Cheng, N. Zheng, and C. Sun, "Boosted Gabor features applied to vehicle detection," *Proc. of IEEE International Conference on Pattern Recognition*, pp. 662-666, 2006.

[5] W. Zheng and L. Liang, "Fast car detection using image strip features," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2703-2710, 2009.

[6] X. Jin and C. H. Davis, "Vehicle detection from high-resolution satellite imagery using morphological shared-weight neural networks," *Image and Vision Computing*, vol. 25, no. 9, pp. 1422-1431, 2007.

[7] B. Wu and R. Nevatia, "Cluster boosted tree classifier for multi-view, multi-pose object detection," *Proc. of IEEE International Conference on Computer Vision*, pp. 1-8, 2007.

[8] H. Y. Cheng, C. C. Weng, and Y. Y. Chen, "Vehicle detection in aerial surveillance using dynamic Bayesian networks," *IEEE Trans. on Image Processing*, vol. 21, no. 4, pp. 2152-2159, 2012.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886-893, 2005.

[10] C. Huang, H. Ai, Y. Li, and S. Lao, "Vector boosting for rotation invariant multi-view face detection," *Proc. of IEEE International Conference on Computer Vision*, pp. 446-453, 2005.

[11] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504-507, 2006.

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

[13] P. Xu, M. Ye, Q. Liu, X. Li, L. Pei, and J. Ding, "Motion detection via a couple of auto-encoder networks," *Proc. of IEEE International Conference on Multimedia and Expo*, pp, 1-6, 2014.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, pp. 2278-2324, 1998.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Proc. of Advances in Neural Information Processing Systems*, pp. 4, 2012.

[16] C. Garcia and M. Delakis, "Convolutional face finder: A neural architecture for fast and robust face detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1408-1423, 2004.

[17] Y. Chen, C. Han, C. Wang, B. Jeng, and K. Fan, "A CNN-based face detector with a simple feature map and a coarse-to-fine classifier," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1-13, 2010.

[18] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3626-3633, 2013.

[19] J. Pang, Q. Huang, S. Yan, S. Jiang, and L. Qin, "Transferring boosted detectors towards viewpoint and scene adaptiveness," *IEEE Trans. on Image Processing*, vol. 20, no. 5, pp. 1388-1400, 2011.

[20] M. Wang, W. Li, and X. Wang, "Transferring a generic pedestrian detector towards specific scenes," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3274-3281, 2012.

[21] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," *Proc. of IEEE International Conference on Computer Vision*, pp. 2146-2153, 2009.

[22] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 511-518, 2001.

[23] H. J. Seo and P. Milanfar, "Training-free, generic object detection using locally adaptive regression kernels," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1688-1704, 2010.

[24] T. L. Saaty, *Analytic Hierarchy Process*, Springer, 2013.

[25] F. F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594-611, 2006.

[26] Caltech computational vision Caltech cars 1999. [Online]. Available: http://www.vision.caltech.edu/html-files/archive.html.

[27] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1475-1490, 2004.

[28] X. Wang, X. Ma, and W. E. L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 539-555, 2009.

[29] P. Carbonetto, G. Dorkó, C. Schmid, H. Kück, and N. De Freitas, "Learning to recognize objects with little supervision," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 219-237, 2008.

[30] P. Xu, M. Ye, X. Li, L. Pei, and P. Jiao, "Object detection using voting spaces trained by few samples," *Optical Engineering*, vol. 52, no. 9, pp. 093105-093105, 2013.

[31] X. Yang, H. Liu, and L. J. Latecki, "Contour-based object detection as dominant set computation," *Pattern Recognition*, vol. 45, no. 5, pp. 1927-1936, 2012.

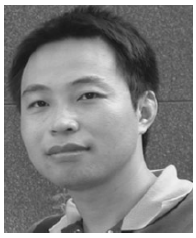
[32] T. Li, M. Ye, and J. Ding, "Discriminative Hough context model for object detection," *The Visual Computer*, vol. 30, no. 1, pp. 59-69, 2014.

[33] C.-C. R. Wang and J.-J. Lien, "Automatic vehicle detection using local features—a statistical approach," *IEEE Trans. on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 83-96, 2008.

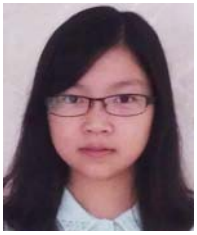
[34] A. Kapoor and J. Winn, "Located hidden random

fields: Learning discriminative parts for object detection," *Proc. of European Conference on Computer Vision*, pp. 302-315, 2006.
[35] B. Wu and R. Nevatia, "Simultaneous object detection and segmentation by boosting local shape feature based classifier," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.
[36] T. Li, M. Ye, F. Pang, H. Y. Wang, and J. Ding, "An efficient fire detection method based on orientation feature," *International Journal of Control, Automation, and Systems*, vol. 11, no. 5, pp. 1038-1045, 2013.

**Xudong Li** received his B.S. degree in Mathematics from Chengdu University of Technology, Chengdu, China, in 2011. He has been taking the successive master-doctor program since September 2011. He is currently a Ph.D. student in University of Electronic Science and Technology of China, Chengdu, China. His current research interests include machine learning and computer vision.



**Mao Ye** received his Ph.D. degree in Mathematics from Chinese University of Hong Kong, in 2002. He is currently a professor and Director of CVLab at University of Electronic Science and Technology of China. His current research interests include machine learning and computer vision. In these areas, he has published over 70 papers in leading international journals or conference proceedings.



**Min Fu** received her bachelor degree from Southwest University of Science and Technology, Mianyang, China, in 2011. As a graduate student she engaged in machine learning and image processing at University of Electronic Science and Technology of China, Chengdu, China and she received her master degree in 2014.

**Pei Xu** received his B.S. degree in Computer Science and Technology from Si-Chuan University of Science and Engineering, ZiGong, China, in 2008 and his MS degree in condensed matter physics from University of Electronic Science and Technology of China, Chengdu, China, in 2011. He is currently a Ph.D. student in University of Electronic Science and Technology of China, Chengdu, China. His current research interests include machine learning and computer vision.

**Tao Li** received his M.E. degree from Central South University, Changsha, China in 2006. He is now a Ph.D. student in University of Electronic Science and Technology, Chengdu, China. His current research interests are machine vision, visual surveillance and object detection.