Coupled Multivehicle Detection and Classification With Prior Objectness Measure

Yanjie Yao, Bin Tian, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—Vehicle recognition plays an important role in traffic surveillance systems, advanced driver-assistance systems, and autonomous vehicles. This paper presents a novel approach for multivehicle recognition that considers vehicle space location and classification as a coupled optimization problem. It can speed up the detection process with more accurate vehicle region proposals and can recognize multiple vehicles using a single model. The proposed detector is implemented in three stages: 1) obtaining candidate vehicle locations with prior objectness measure; 2) classifying vehicle region proposals to distinguish the three common types of vehicles (i.e., car, taxi, and bus) by a single convolutional neural network (CNN); and 3) coupling classification results with the detection process, which leads to fewer false positives. In experiments on high-resolution traffic images, our method achieves unique characteristics: 1) It matches the state-of-the-art detection accuracy; 2) it is more efficient in generating a smaller set of high-quality vehicle windows; 3) its searching time is decreased by about 30 times compared with the other two popular detection schemes; and 4) it recognizes different vehicles in each image using a single CNN model with eight layers.**

***Index Terms*—Convolutional neural network (CNN), multivehicle detection, object proposals, vehicle classification.**

I. Introduction

THE development of intelligent transportation systems (ITS) brings new technologies to solve traffic issues, including congestion, accidents, delays, and pollution. In the applications of ITS, such as traffic light control and intelligent vehicles, there is an increasing demand for traffic data extraction. To extract traffic data automatically and timely, vision-based vehicle recognition is an essential and challenging task. It collects vehicle physical attributes and vehicle traveling data for traffic management and control in parallel transportation systems [1] and has high industrial potential in advanced driver assistance systems [2] and autonomous vehicles [3].

There are two main tasks in typical automated vehicle recognition (AVR) systems: finding locations of vehicles in natural scene images (*vehicle detection*) and classifying detected vehicles into their specific subclasses (*vehicle classification*). According to their recognition feature, different AVR systems have different functions, such as vehicle color recognition systems, vehicle brand recognition systems, and vehicle type recognition (VTR) systems. However, environments of traffic surveillance pose many difficulties for identifying vehicles due to viewpoint variation, multiscale, deformation, illumination conditions, cluttered background, partial occlusion, and motion blur.

To achieve AVR systems, many approaches have been proposed to deal with vehicle detection and classification. In [4], a hierarchical vehicle model was established for real-time vehicle color identification and that could recognize four colors (red/green/blue/yellow) of cars with the help of a support vector machine (SVM) classifier. Lu *et al.* [5] combined the background subtraction method and three frame differencing methods to detect moving vehicles and then classified the detected vehicles into five types by six geometric parameters. Similarly, a VTR system was designed in [6] for a toll station using background subtraction to get vehicles in the region of interest (ROI). Different from the work in [5], it yields vehicle type results by counting the black pixel number included in the vehicle body contour. However, some limitations could be noted in these approaches: 1) Color may dramatically vary in response to illumination changes, and certain color types are very close to other color types; 2) motion-based detection methods are not suitable for slow-moving traffic or car fleet; 3) simple geometric information or pixel counting is not enough to represent a vehicle; and 4) no generic model was proposed for multiple-vehicle detection and classification. Other methods [7]–[9] are based on handcrafted features and complex models, using a category-specific classifier to evaluate image windows in a sliding window fashion. Due to large computation complexity, they are difficult to apply in real-time applications.

In this paper, we propose a deep-learning-based method to recognize multivehicle types in images for traffic surveillance. By considering vehicle space location and classification as a coupled optimization problem, we combine the prior objectness measure [10] and the convolutional neural network (CNN) [11] to recognize multiple vehicles. The main contributions of our work are as follows.

1) We propose a combined probabilistic measure in a Bayesian framework with three cues to help in the search for vehicle locations using objectness scores, which can greatly reduce the number of candidate locations and detection time than with the sliding window technique.

Manuscript received January 6, 2016; revised May 13, 2016; accepted June 14, 2016. Date of publication June 21, 2016; date of current version March 10, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61304200, Grant 61503380, Grant 71232006, and Grant 61233001 and in part by the Natural Science Foundation of Guangdong Province, China, under Grant 2015A030310187. The review of this paper was coordinated by Prof. X. Fang. *(Corresponding author: Bin Tian.)*

The authors are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yanjie.yao@ia.ac.cn; bin.tian@ia.ac.cn; feiyue@ieee.org).

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

Digital Object Identifier 10.1109/TVT.2016.2582926

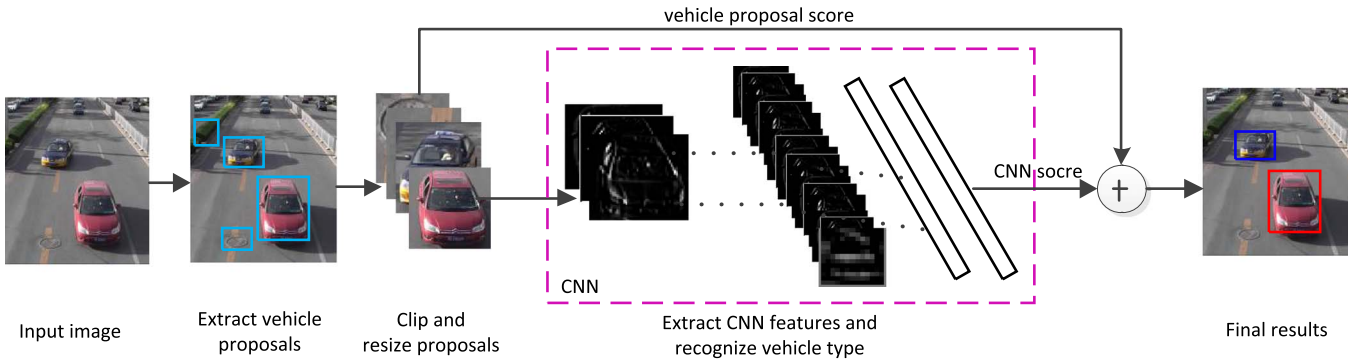


Fig. 1. Pipeline of the proposed VTR algorithm.

2) To recognize different vehicle types in one image, we utilize a CNN that contains eight layers to learn features of vehicle proposals and obtain the corresponding distribution over types with a softmax classifier.
3) To reduce the number of false positives, we linearly combine the type score of a window with its objectness score, which optimizes detection results and classification results simultaneously.

The pipeline of our VTR algorithm is shown in Fig. 1. First, in natural scene images, we carry out the objectness measure to detect vehicle proposals using a Bayesian probability model. Our objectness measure integrates multiple features including multiscale saliency (MS), color contrast (CC), and edge density (ED) to describe the vehicle features more accurately. Then, we sample high score windows in diverse locations by a given window number. Third, we warp the set of candidate detections into a fixed size as a form compatible with the CNN. Then, we further classify the detected vehicles into their specific subclasses by employing a CNN with a softmax classifier. When it ends, a vehicle is recognized as a car, a bus, or a taxi by a linear combination of the proposal score and the CNN score. Experimental results show that both our detection and classification methods achieve a state-of-the-art performance together with significantly improved computational efficiency. It is worth mentioning that our recognition method is 30 times faster than the other two popular detection schemes.

The remainder of this paper is organized as follows. Existing vehicle detection and classification algorithms are generally reviewed in Section II. Our vehicle detection process to extract vehicle proposals is described in Section III. Then, Section IV presents the multivehicle type recognition algorithm and explains the idea of combining the detection and classification results for multiple vehicles. In Section V, the performance of our algorithm is evaluated by real traffic images of metropolitan roads. Experiment results and their comprehensive discussions are also included in the same section. Finally, we make a conclusion of this paper and present the future work.

II. Related Work

Vehicle detection and classification are two basic tasks in vehicle recognition. Here, the existing methods for these two tasks are individually introduced.

A. Vehicle Detection

Vehicle detection requires that the hypothesized locations of vehicles are found and verified quickly in an image [12]. After vehicle detection, further processing can be carried out [13], such as vehicle tracking and vehicle classification. There are two main categories for vehicle detection methods. One is moving-vehicle detection based on background estimation [14], [15]. Vehicle candidates can be found from foreground blocks that are obtained by subtracting the estimated background from original input images. This kind of method has low computational complexity and can be used for applications with a simple and stable background. However, they are not suitable for dealing with congested urban traffic because the congestion causes slow-moving traffic and the lack of movement information. On the other hand, whether an object is moving cannot be determined without considering its inherent information.

Sliding windows [16], [17] is another method for vehicle detection, which is treated as a binary classification problem to distinguish vehicles of different colors and shapes from cluttered backgrounds. The process is as follows: First, parse the whole image with multiscale sliding windows or parse an image pyramid with a fixed sliding window; then, score each sliding window with a classifier based on statistical models to determine whether it contains a vehicle instance or background; and finally, output windows with locally highest scores. The principle is intuitive, and a good detection performance can be got when using proper models and scales.

However, the sliding window mechanism has two potential limitations. First, the sliding window fashion is time consuming, which makes it difficult to be integrated into real-time applications. Parsing the whole image needs millions of windows under different scales, and a larger image yields more windows. When using complex vehicle models such as the deformable part model [18], [19], scoring all windows will cost intolerable computational time. On the other hand, most windows are backgrounds, and it is not necessary to evaluate every window. Second, simply treating vehicle detection as a two-class problem will not satisfy the requirements of vehicle recognition in modern traffic monitoring systems. In reality, multitype vehicles will appear in one image at the same time. Considering all vehicles as the same category cannot describe the details of each vehicle. Although class-specific models can be trained to detect special types of vehicles, such as a taxi [9],

two-classification could not identify which vehicle is a car and which is a taxi by a single model.

To speed up sliding window operations, training an objectness measure [10], [20]–[23], which is generic over categories, has recently become popular for object detection. By proposing a small number of category-independent proposals, the objectness measure, which reflects how likely an image window covers an object, can avoid making decisions early on [10]. Carreira and Sminchisescu [20] and Endres and Hoiem [21] presented effective works on reducing search spaces for classifiers by producing rough segmentations as object proposals, while allowing the usage of strong classifiers to improve accuracy. However, these methods are computationally expensive, usually requiring several minutes per image. In [22], a selective search approach was proposed to get higher prediction performance and was successfully used in regions with a CNN (R-CNN) [24], which is the state-of-the-art object detector. However, the computation cost is still a problem for its real application. For example, when testing a 480 $\times$ 360 pixel image in Caffe [25], 1570 windows are processed in 120 s with a single NVIDIA GTX Titan GPU. In [23], a cascaded ranking SVM approach with an orientated gradient feature was proposed for efficient proposal generation. In [10], a cue integration approach is proposed to get a better prediction performance more efficiently. Inspired by their work, we propose a combined probabilistic objectness measure in a Bayesian framework with three cues to extract multiscale regions as vehicle proposals.

B. Vehicle Classification

Vehicle classification is to classify all detected vehicles into their specific subclasses. Kafai and Bhanu [26] designed a hybrid dynamic Bayesian network that classifies a vehicle as a sedan, a pickup truck, an SUV, or unknown by its height, width, and angle. Chen *et al.* [27] used size and shape cues obtained by camera calibration to classify a vehicle into four classes (car, van, bus, and motorcycle). However, these approaches have a relatively high false-positive rate since they have not considered the appearance or structure features of vehicles, and their performance is heavily influenced by cluttered background, various illuminations, and severe occlusions. In [28], Mishra and Banerjee presented a multifeature combination approach to classify vehicles using SVM. A vehicle is classified to be a two-wheeler, a three-wheeler, a light motor vehicle, or a heavy motor vehicle according to multiple features including Haar, gradient, RGB, and pyramidal histogram of oriented gradients. Unfortunately, selecting and designing an effective handcrafted feature is laborious, and the resulting classifiers are not strong enough to capture vehicles of different poses and scales.

With advances in deep learning and GPU computation, deep CNNs have recently had a major impact in a variety of vision tasks, such as face recognition [29], [30], object detection [24], [31], and object classification [32], [33]. CNNs are biologically inspired multistage architectures that automatically learn hierarchies of invariant features. With its fast development, CNNs are also gradually used in traffic monitoring systems, particularly for traffic sign classification. In [34], a two-stage convolutional network was applied to deal with traffic sign classification for the German Traffic Sign Recognition Benchmark competition [36], which was above the human performance of 98.81% by 98.97% accuracy. In [35], a CNN was used to further classify the detected sign proposals extracted by the color probability model, which was 20 times faster than other existing best traffic sign detection modules.

To adopt the advantages of the CNN, we apply it in solving multiple-vehicle recognition in a real-traffic scene in this paper. We aim at designing a method, which is able to reduce the number of classifier evaluations substantially, detect more precise candidate locations, and recognize multitype vehicles with high accuracy. To achieve the aforementioned idea, a combined probabilistic measure built in a Bayesian framework with three cues is defined to predict a set of bounding boxes, which represent potential vehicle locations. Furthermore, a CNN model is trained to output a score for each box, which indicates whether a specific vehicle type is contained in this box. Here, a candidate box can be classified as a car, a bus, a taxi, or background. Finally, the proposal score and the CNN score are linearly combined for one window, which optimizes detection results and classification results simultaneously and reduces the number of false positives. The details of the method are given in the following sections.

III. Vehicle Proposal Extraction

To extract vehicle proposals, we take the idea of objectness measure to find candidate regions. Objectness is usually represented as a value to quantify how likely an image window covers an object of any class, which can speed up detectors by reducing a large number of evaluated windows. To define the objectness measure, objects in an image are characterized by their uniqueness, a closed boundary in space, and a different appearance from their immediate surroundings. In our work, three image cues are used to measure the characteristics of objects, and the final measure combines them in a Bayesian framework to obtain potential vehicle locations.

A. Three Cues

Alexe *et al.* presented five objectness cues to measure the characteristics for an image window in [10]. In this paper, three of them are selected to get our objectness score. The following gives a brief introduction of them.

Multiscale Saliency: This cue measures the uniqueness characteristic of vehicles. It can measure the unique appearance of a vehicle from backgrounds shown in Fig. 2. For each scale s, a saliency map $M_s(p)$ of an image i at each pixel p can be obtained by the spectral residual of fast Fourier transform proposed in [37]. Extending it to multiple scales, the saliency of a window w at scale s is defined as follows:

$$\mathrm{MS}(w,\theta_s) = \sum_{\{p\in w|M_s(p)\geq\theta_s\}} M_s(p) \times \frac{|\{p\in w|M_s(p)\geq\theta_s\}|}{|w|} \tag{1}$$

where θ_s is scale-specific thresholds, and $|\cdot|$ indicates the number of pixels.

Having MS maps is important for finding more vehicles in data sets. Each scale threshold θ_s is learned independently, by

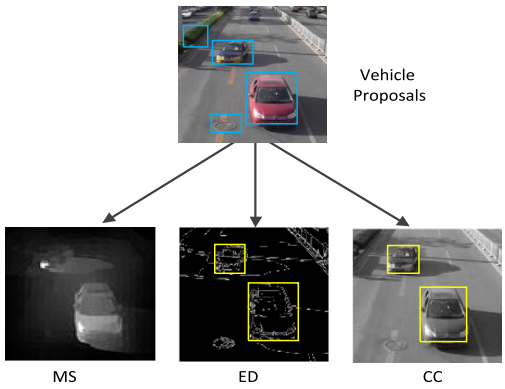


Fig. 2. Bayesian framework combined cues to search for vehicle proposals.

optimizing the localization accuracy of the training vehicle windows V at each scale s. The saliency map $M_s(p)$ and the MS score of all windows are computed for every training image i and scale s. Then, a set of local maxima windows $W^s_{\max}$ is obtained after nonmaximum suppression (NMS) on score space. The optimal θ^*_s is founded by maximizing the following function:

$$\theta^*_s = \arg\max_{\theta_s} \sum\nolimits_{v\in V} \max\nolimits_{w\in W^s_{\max}} \frac{|w\cap v|}{|w\cup v|} \tag{2}$$

the optimal threshold θ^*_s leads the local maxima of MS in the images that can most accurately cover the annotated vehicles. At the same time, maximizing (2) indicates minimizing the score of windows not containing any annotated vehicle.

Edge Density: The ED cue captures the closed-boundary characteristic of vehicles by measuring the density of edges near the window borders. A pixel p that is classified as edge by an edge detector is an edgel. The ED of window w is computed as the density of edgels in the inner ring $In(w,\theta_e)$, i.e.,

$$\mathrm{ED}(w,\theta_e) = \frac{\sum_{p\in In(w,\theta_e)} M_e(p)}{\mathrm{Len}\left(In(w,\theta_e)\right)} \tag{3}$$

where $M_e(p)\in\{0,1\}$ is a binary edge map that is obtained using the Canny detector in this paper, $\mathrm{Len}(\cdot)$ indicates the perimeter of the inner ring, and the inner ring $In(w,\theta_e)$ of window w is obtained by shrinking it by a factor θ_e in all directions, i.e., $|In)w,\theta_e)| = (1/\theta_e^2)|w|$.

The optimal inner ring $In(w,\theta_e)$ is defined by a well-learned parameter θ^*_e. We learn θ_e in a Bayesian framework. For every image i, 100 000 random windows are generated to distinguish positive examples and the negatives. Windows covering an annotated vehicle are considered as positive examples W^{fg}; however, the others are the negatives W^{bg}. For any θ_e, the likelihoods for positive and negative classes can be built as $p(\mathrm{ED}(w,\theta)|fg)$ and $p(\mathrm{ED}(w,\theta)|bg)$, respectively.

The optimal θ^*_e is founded by maximizing the posterior probability that object windows are classified as positives, i.e.,

$$\begin{aligned}\theta^*_e &= \arg\max_{\theta_e} \prod\nolimits_{w\in W^{fg}} p\left(fg|\mathrm{ED}(w,\theta_e)\right)\\ &= \arg\max_{\theta_e} \prod\nolimits_{w\in W^{fg}} \frac{p\left(\mathrm{ED}(w,\theta_e)|fg\right)\cdot p(fg)}{\sum_{c\in\{fg,bg\}} p\left(\mathrm{ED}(w,\theta_e)|c\right)\cdot p(c)}\end{aligned} \tag{4}$$

where the priors are set by relative frequency, i.e.,

$$\begin{cases} p(fg) = \frac{|W^{fg}|}{|W^{fg}|+|W^{bg}|} \\ p(bg) = 1-p(fg). \end{cases} \tag{5}$$

Color Contrast: CC is a useful cue to measure the different appearance characteristics of vehicles. It scores a whole window as whether it contains an entire object. Knowing that objects tend to have a different appearance than the background behind them, CC measures the dissimilarity of a window to its immediate surrounding area according to their color distribution. CC between window w and its surrounding $S(w,\theta_c)$ is computed as

$$\mathrm{CC}(w,\theta_c) = \chi^2\left(h(w), h\left(S(w,\theta_c)\right)\right) \tag{6}$$

where $h(\cdot)$ is the LAB histogram that is invariant to rotation and scales, $\chi^2(\cdot)$ indicates the chi-square distance between two histograms, and the surrounding $S(w,\theta_c)$ of window w is a rectangular ring obtained by enlarging the window by a factor θ_c in all directions, i.e., $|S(w,\theta_c)| = (\theta_c^2-1)|w|$.

Parameter θ_c is learned the same as parameter θe. Note that the learned parameter θ^*_c defines the optimal outer ring $S(w,\theta_c)$. Once all of the parameters have been learned, we can take advantage of the three cues for vehicle proposal detection.

B. Vehicle Proposal Extraction

From the previous section, a vehicle proposal can be measured from backgrounds by its characteristics of uniqueness, closed boundary, and different appearance according to MS, ED, and CC, respectively. To speed up, all cues are computed by integral images. Since the proposed cues are complementary, we combine them in a Bayesian framework to obtain potential vehicle locations in Fig. 2.

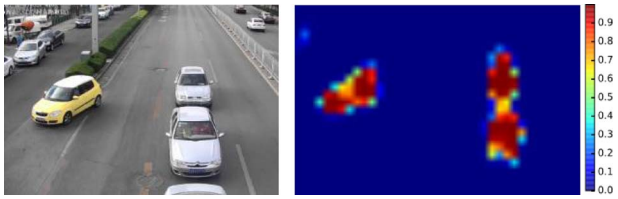
To combine three cues, a Bayesian classifier is trained to distinguish positive from negative. For each training image i, we sample 100 000 windows from the distribution given by the MS cue and then compute the other two cues. The positive and negative examples are similarly defined as in ED. Here, a naive Bayes approach is chosen to avoid enormous samples estimating the joint likelihood of cues.

In our naïve Bayes model, the priors $p(fg)$ and $p(bg)$ can be estimated by (5). Moreover, the individual cue likelihoods $p(\mathrm{cue}|fg)$ and $p(\mathrm{cue}|bg)$ can be obtained due to the fact that cues are independent, where $\mathrm{cue}\in\{\mathrm{SM},\mathrm{ED},\mathrm{CC}\}$. When a test image is given, the posterior probability of a test window w is computed as

$$\begin{aligned} p(fg|\mathrm{cue}) &= \frac{p(c|fg)p(fg)}{p(C)} \\ &= \frac{p(fg)\prod_{\mathrm{cue}} p(\mathrm{cue}|fg)}{\sum_{c\in\{fg,bg\}} p(c)\prod_{\mathrm{cue}} p(\mathrm{cue}|c)}. \end{aligned} \tag{7}$$

Thus, the final objectness score of w is computed by (7).

To get more precise vehicle proposals, we have taken two procedures into account. First, we sample much less candidate vehicle locations according to the desired final number



(a) (b)

Fig. 3. Example of objectness measure to detect vehicle locations. (a) Input image. (b) Corresponding probability heat map of the vehicles' locations.

of windows responding to an objectness threshold. This can reduce a large number of evaluated windows. The selection principle for the window number is described in Section V-A. Then, we consider the size and aspect ratio of the candidate region, which also helps in reducing the false positives. As a complementary strategy, windows that appear too large are reduced by vehicle size prior without analyzing image pixels, such as 500×500. At the same time, a very elongated window is less probable as a vehicle proposal in an image than a square window; hence, this window is also not considered as a vehicle candidate for postprocessing. Fig. 3 gives an example showing how the Bayesian classifier based on the objectness measure can provide the meaningful distribution over the vehicles' locations. Fig. 3(b) is the corresponding probability heat map of an input image that indicates where vehicles are more likely to appear. It proves that our detection procedure can reduce the uncertainty of vehicle locations, which helps us find candidate vehicles quickly and easily.

IV. Vehicle Type Classification Based on Convolutional Neural Network

Here, we will provide details of the vehicle type classification algorithm and its training process by a pretrained CNN. The CNN architecture is trained using the training examples, and later, it acts as a feature extractor to compute a feature vector for each resized image. A softmax classifier over four classes is used to predict the type of a given proposal. As a complementary optimization strategy, a linear combination of the CNN score and the objectness score for a window is used to filter out false positives for final recognition results.

A. Vehicle Type Classification by CNN

Here, a region proposal obtained in Section III can be classified as a car, a bus, a taxi, or background by our VTR model based on a CNN. The CNN feature extractor can run on raw pixels to automatically learn a hierarchy of features in a deep stacked structure for a specific task. Meanwhile, it has the ability to extract features that are invariant to translations, rotations, and scale changes. The framework of our CNN net is shown in Fig. 4. A detailed explanation on this figure is given below.

In our method, we adopt AlexNet [33] as a pretrained model for vehicle type classification. AlexNet is an eight-layer convnet that has been successfully trained on the ILSVRC 2012 ImageNet data set [38]. Before fine tuning the model on our data, we model the recognition task as a four-class classification problem containing four predefined labels: car, bus, taxi, and background. Hence, we replace the final layer of AlexNet with a softmax loss function with a four-dimensional output. As presented in Fig. 4, our model consists of eight layers, where the first five layers are convolution layers $\{C1, C2, C3, C4, C5\}$, and the last three layers are fully connected layers $\{f6, f7, f8\}$. A resized proposal is the input of our CNN model. Based on convolving the input image with different filters, several feature maps can be generated in convolution layers. The responses of the filters in each layer are regarded as the features for our task. Each feature map in pooling layers $\{\text{Pooling1}, \text{Pooling2}\}$ is obtained by max pooling performed on the corresponding feature map in previous convolution layers, respectively. Following each convolution layer, contrast normalization, pooling, and nonlinear function are connected to it successively. Following two fully connected layers $\{f6, f7\}$, the final layer $f8$ implements a softmax nonlinear function to give the score of each category in classification, i.e.,

$$f(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{4} \exp(x_i)} \tag{8}$$

where x_i is the ith input of $f8$ that is equal to a linear combination of a 4096-dimensional feature, and $f(x_i)$ is a four-dimensional output corresponding to the number of nodes in $f8$, which can give a probability to predict the class of a vehicle proposal, i.e., car, taxi, bus, or background.

As shown in Fig. 4, the proposals are resized to 227×227 since the input of the CNN should have the same size. Hence, the net takes $227 \times 227 \times 3$ RGB images as input. The sizes of the filter kernel in the five convolution layers are 11×11, 5×5, 3×3, 3×3, and 3×3, respectively. The sizes of the outputs of all the convolution layers are $55 \times 55 \times 55 \times 96$, $27 \times 27 \times 256$, $13 \times 13 \times 384$, $13 \times 13 \times 384$, and $13 \times 13 \times 256$, respectively. The max pooling method is applied to the outputs of $C1$ and $C2$ to reduce the size of the output and, at the same time, shorten the computation cost. The output of $C5$ is fed to the fully connected layers $f6$ and $f7$ to get a long feature vector with the length of 4096. Finally, these extracted feature vectors are used to compute the score of each class by the softmax classifier. Given all scored regions in an image, a greedy NMS is applied to reject a region if its intersection-over-union (IOU) overlap with a higher scoring selected region is lower than a learned threshold.

For fine tuning, we used 100 k iterations of stochastic gradient descent (SGD), momentum of 0.9, weight decay of 0.0005, and base learning rate of 0.001. Note that the learning rate is dropped to one-tenth of the initial rate every 20 k iterations, which allows fine tuning to progress while not clobbering the initialization. We trained our models using SGD with a batch size of 128 examples, where each batch contained 32 positive and 96 negative examples. To generate examples, we manually annotated the type of each vehicle from the data set, which consists of 10 K images with 40 K vehicles. To increase the number of examples, we randomly sampled subwindows of the annotated images. A subwindow is treated as a positive example if it has more than an 80% IOU overlap with the

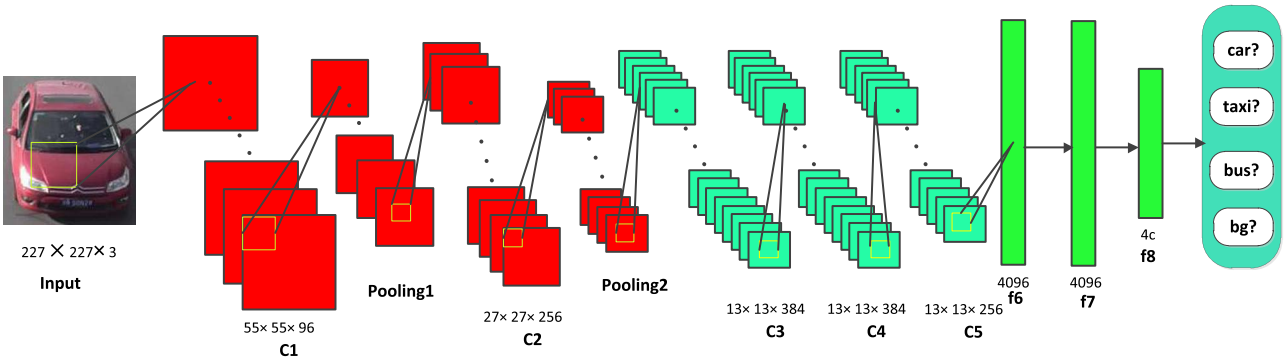


Fig. 4. Framework of our CNN.

Fig. 5. Examples for training our CNN model.

ground-truth box. Otherwise, it is treated as a negative image if it has less than a 20% IOU overlap with the ground-truth box. For further data augmentation, we also cropped and flipped the taxi and bus examples randomly because they are extremely rare compared with car images in real traffic. Fig. 5 shows some examples of training samples. It can be clearly seen that our training samples contain a wide-range rotation angle of vehicles. Finally, the resulted data set contains 30 K positive images and 90 k negative images for training and testing. The fine tuning takes about 9 h in Caffe on a Titan GPU with a very high classification accuracy level of 99.76%.

B. Reduce False-Positive Rate

The score function of the given CNN-based classifier typically returns a high response to instances of vehicle types, but occasionally also to other image patterns, which will usually lead to false positives. The most common false positives are images of full background and partial background with object. To reduce the number of false positives and improve the average precision for three vehicle types, we linearly combine the detection and classification results in the previous sections. A low score of objectness measure should be given to a false-positive window. To realize this, the final score $f(w)$ is calculated by combining the type score $c(W)$ of window w and its objectness score $p(v|w)$, i.e.,

$$f(w) = (1-\alpha)c(w) + \alpha \cdot p(v|w) \tag{9}$$

where α is the weight to control the importance of the objectness score.

When a different value of α is set, the final result for vehicle type classification is also different in some degree. Because that objectness score $p(v|w)$ is an assistant measure to improve the reliability of type score $c(w)$, it is usually set at a smaller weight value. We tested several values for parameter α to obtain better vehicle detection and classification results. In our final experiment, we set $\alpha = 0.2$.

TABLE I
DETAILED INFORMATION OF OUR COLLECTED DATA SET

Dataset	Description	Number of Examples			
		Image	Car	Taxi	Bus
Subset 1	daytime, resolution 1920×1080	300	817	210	106
Subset 2	nighttime, resolution 1920×1080	300	445	139	197
Subset 3	daytime, resolution 2592×1936	300	2214	615	520
Total		900	3476	964	823

V. EXPERIMENTS

We evaluate our integrated approach on a large set of image sequences and compare it with other representative methods. All testing images are taken by traffic cameras along metropolitan roads. All experiments were conducted on a computer with 4-GHz CPU, 32-G RAM, 12-G GPU, and 64-bit Linux OS. Experimental results under various circumstances of roads show that our method achieves the state-of-the-art performance with significantly improved computational efficiency. The recognition process is almost 30 times faster than the R-CNN method.

A. Data Set and Evaluation Criteria

This section presents the data set and evaluation criteria to verify the effectiveness of our method.

The proposed methods are trained and evaluated on a large set of testing images in various traffic conditions including partial occlusion. The images are captured roughly from the frontal view by different high-resolution CCD cameras along metropolitan roads. The data set is built from several videos that are respectively captured at 8 fps with the resolution of 2592 × 1936 and 1920 × 1080. For the training phase, we do some data augmentation to balance different data classes. We subsample 900 testing images to form three representative data sets that cover a large range of variations in view angle and ambient illumination. Detailed information of each data set is shown in Table I. For convenience, the ROI is set from the

TABLE II
PERFORMANCE OF OUR METHOD

Vehicle Type	Vehicle Type Recognition Rate			
	Subset 1	Subset 2	Subset 3	Average
Car	96.73%	91.05%	92.88%	93.55%
Bus	97.64%	92.60%	93.81%	94.68%
Taxi	95.78%	90.15%	92.35%	92.76%

middle to the bottom of each image without considering the upper area because the objects are too small in that area.

The performance of the proposed methods is measured by calculating the detection rate/windows amount (DR-#WIN), the VTR rate, and the receiver operating characteristic (ROC) curve. The computational time of the whole recognition system is also considered.

DR-#WIN means detection rate (DR) given #WIN proposals. This metric is the most popular evaluation criterion for objectness measure methods, where DR is the percentage of ground-truth vehicles covered by selected proposal windows, and #WIN is the number of selected proposal windows. When #WIN is larger, DR is more likely to be higher, but the following processing requires more computing resources. A vehicle is correctly detected only if the percentage of the ground-truth bounding box covered by detected windows is above 0.8. The VTR rate indicates the ability to correctly recognize vehicles of each type.

ROC curves show the performance of different methods with a series of TP–FP (true positive rate and false positive rate) pairs at various threshold settings. The ROC curve of different vehicle types is drawn by adjusting the scoring thresholds in the vehicle localization, as shown in Fig. 9. We tested all data sets in different scenarios to get the summary ROC curve and utilized the least squares method for curve fitting. With the ROC curve, we can choose a relatively good scoring threshold for all scenarios.

B. Experimental Results

From Fig. 3, we obtain the probabilistic response of locations for multiple vehicles by our integrated image cues. To describe its ability to extract vehicle proposals, we compute the DR-#WIN curves of our method on three data set, which is shown in Fig. 6. Different #WIN represents different candidate location numbers. A small set of coarse locations with high DR is sufficient for effective vehicle detection, and it allows complex features to be involved in following processing to achieve better quality and higher efficiency than traditional methods. When WIN = 1000, the DR of our method is already above 96%, which is much higher than using a single cue. It proves that a large size of search space is reduced with little loss of DR for the subsequent VTR. This is the reason for the improved efficiency in our method. It is crucial to obtain the precise bounding box of each vehicle region before recognition. It also indicates that the three cues are complementary and important for finding vehicles in challenging traffic images. Table III shows detailed information on the average processing time in different phases of the proposed method.

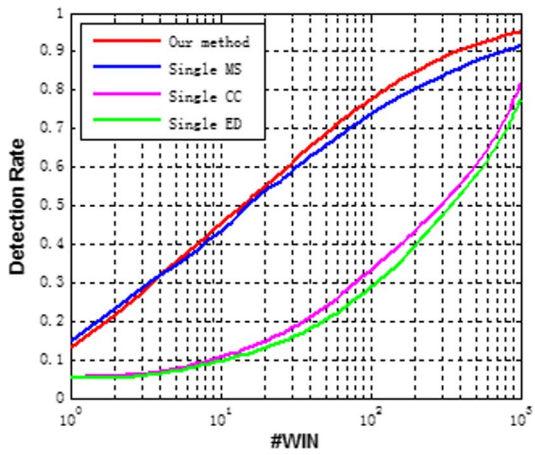


Fig. 6. DR-#WIN plots of the triple cue combinations.

TABLE III
AVERAGE PROCESSING TIME OF DIFFERENT PHASES

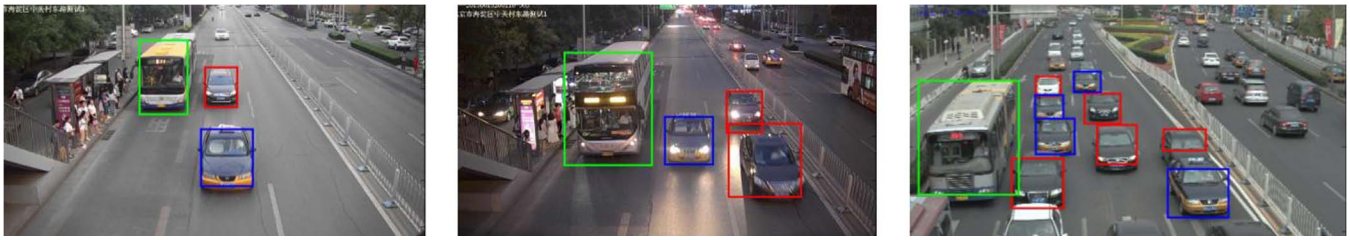
Phases \ Image size	Average Processing Time	
	1920×1080	2592×1936
Vehicle proposals extraction	2.78s	7.52s
Vehicle type classification	2.31s	4.47s

Fig. 7 shows some results of testing images by the proposed method. According to those results, we can easily find that our method can deal with vehicles with different translations, rotations, and noise caused by illuminations. There are two main reasons for this: First, our model is trained on a large-scale data set, which guarantees that the model can be adapted to a variety of situations; second, the robustness depends on multifeature extractors in different stages. The complementary cues and CNN features also ensure that the proposed method can extract features that are invariant to translations, rotations, and noise variances.

Fig. 8 shows examples in the final results of the proposed multivehicle type recognition method. We sequentially process all the testing images and output the bounding boxes of the detected vehicles' type with different colors. The red rectangles indicate the bounding boxes of the detected cars, the blue rectangles indicate the bounding boxes of the detected taxis, and the green rectangles indicate the bounding boxes of the detected buses. As shown in Fig. 8, our method can detect multivehicle locations and recognize the corresponding different vehicle types at the same time. It can work for traffic images under different illumination conditions, including daylight and night. More importantly, our method has a good performance in some occlusion conditions. In Fig. 8(c), it is shown that our method can deal with the partial occlusion between vehicles. In addition, our method adapts to various vehicle poses and shapes benefiting from the usage of the prior objectness measure and the CNN-based classifier. Statistical results are provided in Table II which shows the performance of our method to recognize different vehicle types in each subset. It proves that the proposed method can obtain high precision and meet the requirements of monitoring accuracy.

(a) (b) (c) (d)

Fig. 7. Examples of results for vehicles with different out-of-plane translations, rotations, and illuminations.



(a) (b) (c)

Fig. 8. Examples for detection and recognition results of test images in each subset. (a) Multivehicle type recognition result for a 1920 × 1080 image in the daytime. (b) Multivehicle type recognition result for a 1920 × 1080 image in the nighttime. (c) Multivehicle type recognition result for a 2592 × 1936 image in the daytime.

TABLE IV
COMPARISON OF COMPUTATIONAL TIME FOR THREE METHODS

Image Size	Method	Average Processing Time for an Image
1920×1080	Sliding window with CNN	210s
	R-CNN	167s
	Our method	5s
2592×1936	Sliding window with CNN	450s
	R-CNN	275s
	Our method	12s

C. Comparison of Experiments

Here, some contrasting experiments for further testing of our method have been conducted. The proposed method is compared with two popular object detection schemes, namely, the sliding window technique and the R-CNN method [26]. The sliding window technique is realized by a multiscale pyramid iterative method combining with CNN. The R-CNN method is the state-of-the-art object detection algorithm that adopts selective search, which is another common objectness measure method. The comparison analysis is done from two aspects, namely, computational time and VTR rate.

All of the three methods are conducted in GPU mode. The code is implemented in Python, C++, and MATLAB. Detailed information of the average computational time to process an image for each method is shown in Table IV. Clearly, our method achieves a remarkable advantage in shortening the full computation cost for images in 2 and 5 MP. The sliding window fashion and the selective search method are time consuming, requiring hundreds of seconds to process an image. Our method is efficient in decreasing the processing time for two main reasons: The first reason is that integral images are used to efficiently compute three cues for the final objectness score of a window, and the second reason is that a large number of windows have been reduced before the final evaluation by the objectness measure in detection. As shown in Table IV, the average processing time for different high-resolution traffic images is no more than 20 s in our method. The proposed method is able to efficiently process a 1920 × 1080 image with only 5 s and a 2592 × 1936 image with only 12 s, which is about 30 times faster than the existing R-CNN with selective search. Combined with Fig. 6, it proves that our method can greatly reduce the size of search space without sacrificing the DR.

In Fig. 9, the ROC curves of the three methods for multivehicle type recognition in subset 1 are shown. The curves are color coded so that the proposed method, the R-CNN method, and the sliding window technique appear as red, green, and blue, respectively. The comparison of ROC results clearly shows that our method achieves remarkable advantages on the true positive rate against the same false positive rate above 0.1, for three types of vehicle recognition. This indicates that compared with general R-CNN and sliding windows, our method that couples multivehicle detection and classification is more precise in capturing diverse locations of vehicles and classifying their corresponding types. As shown in Fig. 9, our method has very strong discriminative power and can achieve the state-of-the-art recognition performance for a car, a bus, and a taxi. Our method has shown its advantage in classifying high-dimensional features using a single CNN-based model. In addition, it achieves an effective performance and is robust in dealing with traffic images of different resolution and different illumination conditions.

D. Discussions

The collection of incorrect and missed samples in detection and classification is used to analyze the limitations of our

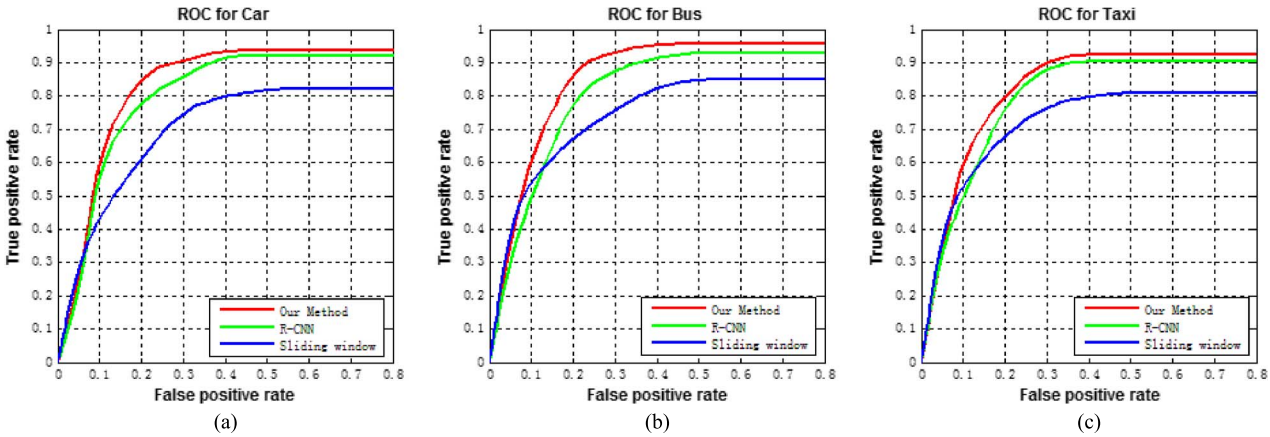


Fig. 9. ROC curves in subset 1 for vehicle types of (a) car, (b) bus, and (c) taxi, respectively.

method. Three situations cause the most failed cases. First, poor image cues caused by the camera view and make vehicles hard to identify. For example, a red taxi can be classified as a car because they are similar in size and color when viewed from one perspective. Second, shadows in daytime and light of vehicles at night cause problems in finding an accurate vehicle location. Thus, the location of a vehicle is vague and generates no strong responses of objectness measure in general. Third, vehicles with severe occlusion are still difficult to detect and classify. In this situation, the objectness score of the occluded hypothesis is quite low, and the occluded is detected as the same vehicle in front by mistake.

VI. Conclusion

A novel method for multivehicle recognition has been proposed in this paper. The proposed method considers vehicle detection and classification as a coupled optimization problem by combining objectness measure with CNN. With three image cues, our approach obtains more accurate vehicle region proposals and avoids the brute-force search in the sliding window approach. Then, normalized detection areas are classified into one of three common vehicle types using a single eight-layer CNN model. Due to the recognition framework, not only are vehicle locations detected, but vehicle types are determined as well. Our method has the ability to extract features that are robust against various translations, rotations, and noise variances. In experiments on high-resolution traffic images, the results have demonstrated that the proposed method can achieve reliable and robust recognition performance in a real-traffic environment while speeding up the detection process by capitalizing on the reduced number of locations.

In addition, the CNN structure makes it suitable for a parallel implementation on GPUs, thus making a real-time recognition system possible. In the future, we are planning to use multiple GPUs to accelerate the vehicle recognition process, improving the performance and efficiency of the recognition system. At the same time, we will also expand the network learning data set and use more sophisticated data augmentation techniques to further recognize more vehicle types and improve our method's performance.

References

[1] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.

[2] Y. C. Kuo, N. S. Pai, and Y. F. Li, "Vision-based vehicle detection for a driver assistance system," *Comput. Math. Appl.*, vol. 61, no. 8, pp. 2096–2100, 2011.

[3] X. Liu, X. Xu, and B. Dai, "Vision-based long-distance lane perception and front vehicle location for full autonomous vehicles on highway roads," *J. Central South Univ.*, vol. 19, pp. 1454–1465, 2012.

[4] C. Y. Lin, C. H. Yeh, and C. H. Yeh, "Real-time vehicle color identification for surveillance videos," in *Proc. IEEE Conf. Electron., Commun. Comput.*, 2014, pp. 59–64.

[5] A. Lu, L. Zhong, L. Li, and Q. Wang, "Moving vehicle recognition and feature extraction from tunnel monitoring videos," *TELKOMNIKA Indonesian J. Elect. Eng.*, vol. 11, no. 10, pp. 6060–6067, 2013.

[6] W. Zhan and Z. Q. Luo, "Research of vehicle type recognition system based on audio video interleaved flow for toll station," *J. Softw.*, vol. 7, no. 4, pp. 741–744, 2012.

[7] H. Cho and S. Y. Hwang, "High-performance on-road vehicle detection with non-biased cascade classifier by weight-balanced training," *J. Image Video Process.*, vol. 2015, no. 16, pp. 1–7, 2015.

[8] Y. Li, B. Tian, B. Li, G. Xiong, F. H. Zhu, and K. F. Wang, "Vehicle detection with a part-based model for complex traffic conditions," in *Proc. IEEE Conf. Veh. Electron. Safety*, 2013, pp. 110–113.

[9] B. Tian, B. Li, Y. Li, G. Xiong, and F. H. Zhu, "Taxi detection based on vehicle painting features for urban traffic scenes," in *Proc. IEEE Conf. Veh. Electron. Safety*, 2013, pp. 105–109.

[10] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

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

[12] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694–711, May 2006.

[13] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2259–2272, Nov. 2011.

[14] J. Zhou, D. Gao, and D. Zhang, "Moving vehicle detection for automatic traffic monitoring," *IEEE Trans. Veh. Technol.*, vol. 56, no. 1, pp. 51–59, Jan. 2007.

[15] M. Vargas, J. M. Milla, S. L. Toral, and F. Barrero, "An enhanced background estimation algorithm for vehicle detection in urban traffic scenes," *IEEE Trans. Veh. Technol.*, vol. 59, no. 8, pp. 3694–3709, Oct. 2010.

[16] B. Tian *et al.*, "Hierarchical and networked vehicle surveillance in ITS: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 557–580, Apr. 2015.

[17] R. Feris *et al.*, "Large-scale vehicle detection, indexing, and search in urban surveillance videos," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 28–42, Feb. 2012.

[18] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[19] L. C. Leon and J. R. Hirata, "Vehicle detection using mixture of deformable parts models: Static and dynamic camera," in *Proc. SIBGRAPI Conf. Graphics, Patterns Images*, 2012, pp. 237–244.
[20] J. Carreira and C. Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric min-cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1312–1328, Jul. 2012.
[21] I. Endres and D. Hoiem, "Category independent object proposals," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 575–588.
[22] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 34, no. 7, pp. 1312–1328, 2013.
[23] Z. Zhang, J. Warrell, and P. H. Torr, "Proposal generation for object detection using cascaded ranking SVMs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1497–1504.
[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 580–587.
[25] Y. Jia, *Caffe: An Open Source Convolutional Architecture for Fast Feature Embedding*, Mar. 2013. [Online]. Available: http://caffe.berkeleyvision.org/
[26] M. Kafai and B. Bhanu, "Dynamic Bayesian networks for vehicle classification in video," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 100–109, Feb. 2012.
[27] Z. Chen, T. Ellis, and S. Velastin, "Vehicle type categorization: A comparison of classification schemes," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Oct. 2011, pp. 74–79.
[28] P. K. Mishra and B. Banerjee, "Vehicle classification using density based multi-feature approach in support vector machine classifier," *Int. J. Comput. Appl.*, vol. 71, no. 7, pp. 1–6, Jun. 2013.
[29] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 1701–1708.
[30] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. NIPS*, 2014, pp. 1988–1996.
[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in Proc. *Int. Conf. Learn. Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6229
[32] R. Socher, B. Huval, B. Bath, D. M. Christopher, and Y. N. Andrew, "Convolutional-recursive deep learning for 3D object classification," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 665–673, 2012.
[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[34] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. IJCNN*, 2011, pp. 2809–2813.
[35] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 2022–2031, 2016.
[36] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *Proc. IJCNN*, 2011, pp. 1453–1460.
[37] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.
[38] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and F.-F. Li, *ImageNet Large Scale Visual Recognition Competition 2012 (ILSVR012)*. [Online]. Available: http://www.image-net.org/challenges/LSVRC/2012/

Yanjie Yao received the B.S. degree from the China University of Geosciences, Beijing, China, in 2011. She is currently working toward the Ph.D. degree in control theory and control engineering with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing.

Her research interests include intelligent transportation systems, image processing, and computer vision.

Bin Tian received the B.S. degree from Shandong University, Jinan, China, in 2009 and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014.

He is currently an Assistant Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include visual object detection, machine learning, and intelligent transportation systems.

Fei-Yue Wang (S'87–M'89–SM'94–F'03) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

In 1990, he joined the University of Arizona, Tucson, AZ, USA, and became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center at the Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Overseas Chinese Talents Program, and in 2002, he was appointed as the Director of the CAS Key Laboratory for Complex Systems and Intelligence Science. From 2006 to 2010, he was the Vice President for Research, Education, and Academic Exchanges with the Institute of Automation, CAS. Since 2005, he has been the Dean of the School of Software Engineering with Xi'an Jiaotong University, Xi'an, China. In 2011, he became the State Specially Appointed Expert and the Founding Director of the State Key Laboratory of Management and Control for Complex Systems. Over the past three decades, he has published over ten books and 300 papers in his areas of interest. His research is focused on social computing and parallel systems.

Dr. Wang was the Editor-in-Chief of the IEEE INTELLIGENT SYSTEMS from 2009 to 2012. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He has served as a General or a Program Chair for more than 20 IEEE, Institute for Operations Research and the Management Sciences, Association for Computing Machinery (ACM), and American Society of Mechanical Engineers (ASME) conferences. He was the President of the IEEE Intelligent Transportation Systems (ITS) Society from 2005 to 2007, the Chinese Association for Science and Technology (USA) in 2005, and the American Zhu Kezhen Education Foundation from 2007 to 2008. He is a member of Sigma Xi, an Outstanding Scientist of ACM, and a Fellow of the International Federation of Automatic Control, the International Council on Systems Engineering, ASME, and the American Association for the Advancement of Science. He is currently the Vice President and Secretary General of the Chinese Association of Automation. In 2007, he received the Second Class National Prize in Natural Sciences of China for his work in intelligent control and social computing. He received the IEEE ITS Outstanding Application Award in 2009 and 2015, the IEEE ITS Outstanding Research Award in 2011, the IEEE Intelligence and Security Informatics Outstanding Research Award in 2012, and the ASME MESA Achievement Award in 2012 for his cumulative contribution to the field of mechatronic/embedded systems and applications. In 2014, he received the IEEE Systems, Man, and Cybernetics Society Norbert Wiener Award.