# Coupled Multi-Vehicle Detection and Classification with Prior Objectness Measure 

Yanjie Yao, Bin Tian, 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 multi-vehicle recognition which 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 multi-vehicles using a single model. The proposed detector is implemented by three stages: 1) Obtaining candidate vehicle locations with prior objectness measure; 2) classifying vehicle region proposals to distinguish three common types of vehicles (i.e. car, taxi, and bus) by a single convolutional neural network; and 3) coupling classification results with detection process which lead 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 efficiently generating smaller set of high quality vehicle windows; 3) searching time is decreased about 30 times compared with other two popular detection schemes; and 4) it recognizes different vehicles in each image using a single CNN model with 8 -layers.


Index Terms-Object proposals, multi-vehicle detection, vehicle classification, convolutional neural network (CNN).

## I. Introduction

THE development of intelligent transportation system (ITS) brings new technologies to solve traffic issues including congestion, accidents, delays, and pollution, etc. In 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].

[^0]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 sub-classes (vehicle classification). According to their recognition feature, different AVR systems have different functions, such as vehicle color recognition (VCR) systems, vehicle brand recognition (VBR) systems, and vehicle type recognition (VTR) systems. However, environments of traffic surveillance pose many difficulties for identifying vehicles due to viewpoint variation, multi-scale, deformation, illumination conditions, cluttered background, partial occlusion, and motion blur, etc.

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 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 background subtraction method and three frame differencing method to detect moving vehicles, then classified detected vehicles into five types by six geometric parameters. Similarly, a VTR system was designed in [6] for toll station using background subtraction to get vehicles in region of interest (ROI). Different from [5], it yields vehicle type results by counting black pixels number included in vehicle body contour. However, some limitations could be noted in these approaches: 1) color may vary dramatically 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 cars fleet; 3) simple geometric information or pixel counting is not enough to represent a vehicle; 4) no generic model was proposed for multiple vehicle detection and classification. Other methods [7]-[9] are based on hand-crafted features and complex models, using category specific classifier to evaluate image windows in a sliding window fashion. Due to large computation complexity, they are difficult to be applied in real-time applications.

In this paper, we propose a deep-learning-based method to recognize multi-vehicle types in images for traffic surveillance. By considering vehicle space location and classification as a coupled optimization problem, we combine prior objectness measure [10] and 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 Bayesian framework with three cues to help the search for


Fig. 1. The pipeline of the proposed vehicle type recognition algorithm.
vehicle locations using objectness scores, which can greatly reduce the number of candidate locations and detection time than sliding windows technique.
(2) To recognize different vehicle types in one image, we utilize a CNN which contains 8 layers to learn features of vehicle proposals, and obtain 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 vehicle type recognition algorithm is illustrated in Fig. 1. Firstly, in natural scene images, we carry out objectness measure to detect vehicle proposals using Bayesian probability model. Our objectness measure integrates multiple features including multi-scale saliency, color contrast, and edge density so as to describe the vehicle features more accurate. Then, we sample high score windows in diverse locations by a given window number. Thirdly, we warp the set of candidate detections into a fixed size as a form compatible with the CNN. After that, we further classify the detected vehicles into their specific sub-classes by employing CNN with softmax classifier. When it ends, a vehicle is recognized as a car, a bus, or a taxi by a linear combination of proposal score and CNN score. Experimental results show that both our detection and classification methods achieve state-of-the-art performance together with significantly improved computational efficiency. It is worth mentioning that our recognition method is 30 times faster than 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 multi-vehicle type recognition algorithm, and explains the idea of combining the detection and classification results for multi-vehicle. 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. In this section, existing methods for these two tasks are introduced individually.

## 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 which are obtained by subtracting estimated background from original input images. This kind of method has low computational complexity, and can be used for applications with simple and stable background. However, they are not suitable to deal with congested urban traffic because the congestion causes slow-moving traffic and the lack of movement information. On the other hand, an object can't only be determined by whether it is moving 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: parse the whole image with multi-scale sliding windows or parse an image pyramid with a fixed sliding window, 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 good detection performance can be got when using proper models and scales.

However, the sliding window mechanism has two potential limitations. Firstly, the sliding window fashion is time consuming which makes it difficult to be integrated into real-time applications. Parsing the whole image needs millions windows under different scales, and larger image yields more windows. When using complex vehicle models such as deformable Parts Model [18][19], scoring all windows will cost intolerable computational time. Whereas, most windows are backgrounds and it is not necessary to evaluate every window. Secondly, simply treating vehicle detection as a two-class problem can't satisfy requirements of vehicle recognition in modern traffic monitoring systems. In reality, multi-type vehicles will appear in one image at the same time.

Considering all vehicles as the same category can't describe the details of each vehicle. Though class-specific models can be trained to detect special type of vehicles, such as 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, objectness measure, which reflects how likely an image window covers an object, can avoid making decisions early on [10]. Carreira et al. [20] and Endres et al. [21] presented effective works of reducing search spaces for classifiers by producing rough segmentations as object proposals, whilst 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 CNN (R-CNN) [24], which is the state-of-the-art object detector. But computation cost is still a problem for its real application. For example, when testing a $480 \times 360$ pixels image in caffe [25], 1570 windows are processed in 120s with a single NVIDIA GTX Titan GPU. In [23], a cascaded ranking SVM approach with orientated gradient feature was proposed for efficient proposal generation. In [10], Alexe et al. proposed a cue integration approach to get better prediction performance more efficiently. Inspired by their work, we propose a combined probabilistic objectness measure in Bayesian framework with three cues to extract multi-scale regions as vehicle proposals.

## B. Vehicle Classification

Vehicle classification is to classify all detected vehicles into their specific sub-classes. Kafai et al. [26] designed a hybrid dynamic Bayesian network which classifies a vehicle into sedan, pickup truck, 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 appearance or structure features of vehicles, and their performance is heavily influenced by cluttered background, various illuminations, and severe occlusions. In [28], the authors presented a multi-feature combination approach to classify vehicles using SVM. A vehicle is classified to be a 2 -wheeler, 3-wheeler, light motor vehicle or 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 resulted classifiers are not strong enough to capture vehicles of different poses and scales.

With advances in deep learning and GPU computation, deep convolutional neural networks (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
multi-stage architectures which automatically learn hierarchies of invariant features. With its fast development, CNNs are also gradually used in traffic monitoring systems, especially for traffic sign classification. In [34], a two-stage Convolutional Networks was applied to deal with traffic sign classification for GTSRB 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 color probability model, which was 20 times faster than other existing best traffic sign detection module.

To adopt the advantages of CNN, we apply it to solve multiple-vehicle recognition in 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 multi-type vehicles with high accuracy. To achieve the above idea, a combined probabilistic measure built in Bayesian framework with three cues is defined to predict a set of bounding boxes, which represent potential vehicle locations. Furtherly, a CNN model is trained to output a score for each box which indicates whether specific vehicle type is contained in this box. Here, a candidate box can be classified into a car, bus, taxi, or background. Finally, proposal score and 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 Proposals 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 respectively, 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.
Multi-scale Saliency (MS): This cue measures the uniqueness characteristic of vehicles. It can measure 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 Transformation (FFT) proposed in [37]. Extending it to multiple scales, the saliency of a window $w$ at scale $s$ is defined as follows

$$
\operatorname{MS}\left(w, \theta_{S}\right)=\sum_{\left\{p \in w \mid M_{s}(p) \geq \theta_{s}\right\}} M_{s}(p) \times \frac{\left|\left\{p \in w \mid M_{s}(p) \geq \theta_{s}\right\}\right|}{|w|}(1)
$$



Fig. 2. Bayesian framework combined cues to search for vehicles proposals.
where $\theta_{s}$ is scale-specific thresholds, and $|\cdot|$ indicates the number of pixels.

Having multi-scale saliency maps is important for finding more vehicles in datasets. Each scale threshold $\theta_{s}$ is learned independently, by 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 $W_{\max }^{s}$ is obtained after NMS on score space. The optimal $\theta_{s}^{*}$ is founded by maximizing the following function:

$$
\begin{equation*}
\theta_{s}^{*}=\arg \max _{\theta_{s}} \sum_{v \in V} \max _{w \in W_{\max }^{s}} \frac{|w \cap v|}{|w \cup v|} \tag{2}
\end{equation*}
$$

the optimal threshold $\theta_{s}^{*}$ leads the local maxima of MS in the images which 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 ( $E D$ ): The ED cue captures the closed boundary characteristic of vehicles by measuring the density of edges near the window borders. A pixel $p$ which is classified as edge by an edge detector is an edgel. The ED of a window $w$ is computed as the density of edgels in the inner ring $\operatorname{In}\left(w, \theta_{e}\right)$

$$
\begin{equation*}
\mathrm{ED}\left(\mathrm{w}, \theta_{e}\right)=\frac{\sum_{p \in \operatorname{In}\left(w, \theta_{e}\right)} M_{e}(p)}{\operatorname{Len}\left(\operatorname{In}\left(w, \theta_{e}\right)\right)} \tag{3}
\end{equation*}
$$

where $M_{e}(p) \in\{0,1\}$ is a binary edge map which is obtained using the Canny detector in this paper, Len (•) indicates the perimeter of the inner ring, and the inner ring $\operatorname{In}\left(w, \theta_{e}\right)$ of a window w is obtained by shrinking it by a factor $\theta_{e}$ in all directions, i.e. $\left|\operatorname{In}\left(\mathrm{w}, \theta_{e}\right)\right|=\frac{1}{\theta_{e}^{2}}|w|$.

The optimal inner ring $\operatorname{In}\left(\mathrm{w}, \theta_{e}\right)$ is defined by a well-learned parameter $\theta_{e}^{*}$. We learn $\theta_{e}$ in a Bayesian framework. For every image $i, 100000$ random windows are generated to distinguish positive examples and the negatives. Windows covering an annotated vehicle are considered as positive examples $W^{f g}$, yet the others are the negatives $W^{b g}$. For any $\theta_{e}$, the likelihoods for positive and negative classes can be built as $p(E D(w, \theta) \mid f g)$ and $p(E D(w, \theta) \mid b g)$, respectively.

The optimal $\theta_{e}^{*}$ is founded by maximizing the posterior probability that object windows are classified as positives:

$$
\begin{aligned}
\theta_{e}^{*} & =\arg \max _{\theta_{e}} \Pi_{w \in W^{f g}} p\left(\mathrm{fg} \mid \mathrm{ED}\left(\mathrm{w}, \theta_{e}\right)\right) \\
& =\arg \max _{\theta_{e}} \Pi_{w \in W^{f g}} \frac{p\left(\mathrm{ED}\left(\mathrm{w}, \theta_{e}\right) \mid f g\right) \cdot p(f g)}{\sum_{c \in\{f g, b g\}} p\left(\mathrm{ED}\left(\mathrm{w}, \theta_{e}\right) \mid c\right) \cdot p(c)}
\end{aligned}
$$

where the priors are set by relative frequency:

$$
\left\{\begin{array}{l}
p(\mathrm{fg})=\frac{\left|W^{f g}\right|}{\left|W^{f g}\right|+\mid W^{b g \mid}}  \tag{5}\\
p(\mathrm{bg})=1-p(f g)
\end{array}\right.
$$

Color Contrast (CC): CC is a useful cue to measure the different appearance characteristic 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 a window $w$ and its surrounding $S\left(w, \theta_{c}\right)$ is computed as

$$
\begin{equation*}
\operatorname{CC}\left(w, \theta_{c}\right)=\chi^{2}\left(h(w), h\left(S\left(w, \theta_{c}\right)\right)\right) \tag{6}
\end{equation*}
$$

where $h(\cdot)$ is the LAB histogram which is invariant to rotation and scales, $\chi^{2}(\cdot)$ indicates the Chi-square distance between two histograms, and the surrounding $\mathrm{S}\left(\mathrm{w}, \theta_{\mathrm{c}}\right)$ of a window $w$ is a rectangular ring obtained by enlarging the window by a factor $\theta_{c}$ in all directions, i.e. $\left|S\left(w, \theta_{c}\right)\right|=\left(\theta_{c}^{2}-1\right)|w|$.
The parameter $\theta_{c}$ is learned as same as the parameter $\theta_{e}$. Note that the learned parameter $\theta_{c}^{*}$ defines the optimal outer ring $S\left(w, \theta_{c}\right)$. Once all of the parameters have been learned, we can take advantage of the three cues for vehicle proposals detection.

## B. Vehicle Proposals Extraction

From above subsection, 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 100000 windows from the distribution given by the MS cue, and then compute the other two cues. The positive and negative examples are defined in the same way as in ED. Here, a Naive Bayes approach is chosen to avoid enormous samples to estimate the joint likelihood of cues.

In our Naive Bayes model, the priors $p(f g)$ and $p(b g)$ can be estimated by above eq. (5). And the individual cue likelihood $p(c u e \mid f g$ ) and $p(c u e \mid b g)$ can be obtained due to cues are independent, where $c u e \in\{S M, E D, C C\}$. When a test image is given, the posterior probability of a test window $w$ is computed as

$$
\begin{align*}
\mathrm{p}(\mathrm{fg} \mid \text { cue }) & =\frac{\mathrm{p}(\mathrm{c} \mid \mathrm{fg}) \mathrm{p}(\mathrm{fg})}{\mathrm{p}(\mathrm{c})}  \tag{7}\\
= & \frac{\mathrm{p}(\mathrm{fg}) \prod_{\text {cue }} \mathrm{p}(\text { cue } \mid \mathrm{fg})}{\sum_{\mathrm{c} \in\{\mathrm{fg}, \mathrm{bg}\}} \mathrm{p}(\mathrm{c}) \prod_{\text {cue }} \mathrm{p}(\mathrm{cue} \mid \mathrm{c})}
\end{align*}
$$

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


Fig. 3. An example of objectness measure to detect vehicle locations. (a) An input image. (b)The corresponding probability heat-map of vehicles' locations.


Fig. 5. Examples for training our CNN model.


Fig. 4. The framework of our CNN

To get more precise vehicle proposals, we have taken two procedures into account. Firstly, we sample much less candidate vehicle locations according to desired final number of windows responding to an objectness threshold. This can reduce a large number of evaluated windows. The selection principle for windows number is described in Section V-A. After that, we consider the size and aspect ratio of the candidate region which also helps 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 $500 \times 500$. At the same time, a very elongated window is less probable as a vehicle proposal in an image than a square window, so this window is also not considered as a vehicle candidate for the postprocessing. Fig. 3 gives an example to show how the Bayesian classifier based on objectness measure can provide the meaningful distribution over 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 CNN

In this section, we will provide details of the vehicle type classification algorithm and its training process by a pre-trained 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 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 into a car, bus, taxi, or background by our vehicle type recognition model based on CNN. The CNN feature extractor can runs 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 which are invariant to translations, rotations, and scale changes. The framework of our CNN net is depicted in Fig. 4. A detailed explaination on this figure is given below.

In our method, we adopt AlexNet [33] as a pre-trained model for vehicle type classification. AlexNet is an 8 layers convnet which has been successfully trained on the ILSVRC 2012 ImageNet dataset [38]. Before fine-tuning the model on our data, we model the recognition task as a 4-class classification problem containing four predefined labels: car, bus, taxi and background. So we replace the final layer of AlexNet with a Softmax loss function with a 4-dimension output. As presented in Fig. 4, our model consists of 8 layers, where the first 5 layers are convolution layers $\{C 1, C 2, C 3, C 4, C 5\}$ and the last 3 layers are fully-connected layers $\{f 6, f 7, f 8\}$. 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 \{Pooling1,Pooling2\} is obtained by max pooling that is performed on the corresponding feature map in previous convolution layers, respectively. Following each convolution layer, the contrast normalization, pooling, and nonlinear function are connected to it successively. Following two fully-connected layers $\{f 6, f 7\}$, the final layer $f 8$ implements a softmax nonlinear function to give the score of each category in classification:

$$
\begin{equation*}
\mathrm{f}\left(x_{i}\right)=\frac{\exp \left(x_{i}\right)}{\sum_{j=1}^{4} \exp \left(x_{i}\right)} \tag{8}
\end{equation*}
$$

where $x_{i}$ is $i_{t h}$ input of $f 8$ that equals to a linear combination of 4096-dimension feature, and $\mathrm{f}\left(x_{i}\right)$ is a 4-dimension output corresponding to the number of nodes in $f 8$ which can give a

TABLE I
DETAILED INFORMATION OF OUR COLLECTED DATASET

| Dataset | Description | Number of Examples |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  | Image | Car | Taxi | Bus |  |
| Subset 1 | daytime, <br> resolution $1920 \times 1080$ | 300 | 817 | 210 | 106 |
| Subset 2 | nighttime, <br> resolution $1920 \times 1080$ | 300 | 445 | 139 | 197 |
| Subset 3 | daytime, <br> resolution $2592 \times 1936$ | 300 | 2214 | 615 | 520 |
|  | Total | 900 | 3476 | 964 | 823 |

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 $227 \times 227$ since the input of CNN should have the same size. So the net takes $227 \times 227 \times 3$ RGB images as input. The size of filter kernel in the five convolution layers are $11 \times 11,5 \times 5,3 \times$ $3,3 \times 3,3 \times 3$, respectively. The sizes of the outputs of all the convolution layers are $55 \times 55 \times 96,27 \times 27 \times 256,13 \times$ $13 \times 384,13 \times 13 \times 384$, and $13 \times 13 \times 256$, respectively. Max pooling method is applied to the outputs of $C 1$ and $C 2$ to reduce the size of the output and shorten the computation cost at the same time. The output of $C 5$ is fed to the fully-connected layers $f 6$ and $f 7$ to get a long feature vector with the length of 4096. Finally, these extracted feature vectors are used to compute 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 $1 / 10$ th of the initial rate every 20 k iterations, which allows fine-tuning to make 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 dataset, 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 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 to 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 dataset contains 30K positive images and 90k negative images for training and testing. The fine-tuning takes about 9 hours in caffe on a Titan GPU with a very high classification accuracy of $99.76 \%$.

## B. Reduce False-Positive Rate

The score function of above CNN-based classifier typically returns a high response to instances of vehicle types, but

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 | $95.78 \%$ | $90.15 \%$ | $92.35 \%$ | $92.76 \%$ |

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. In order 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 above 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 a window $w$ and its objectness score $p(v \mid w)$ :

$$
\begin{equation*}
\mathrm{f}(w)=(1-\alpha) c(w)+\alpha \cdot p(v \mid w) \tag{9}
\end{equation*}
$$

where $\alpha$ is the weight to control the importance of objectness score.

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

## 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, 32G RAM, 12G GPU, and 64bit 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 R-CNN method.

## A. Dataset and Evaluation Criteria

This section presents dataset 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 dataset is built from several videos that are respectively captured at 8 fps with the resolution of $2592 \times 1936$ and $1920 \times 1080$. For training phase, we do some data augmentation to balance different data class. We subsample 900 testing images to form three representative datasets which cover a large range of variations in view angle and ambient illumination. Detailed information of each dataset


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

TABLE III

| AvERAGE PROCESSING TiME OF DIFFERENT PHASES |  |  |
| :--- | :---: | :---: |
|  | Average Processing Time |  |
| Image size | $1920 \times 1080$ | $2592 \times 1936$ |
| Vehicle proposals <br> extraction | 2.78 s | 7.52 s |
| Vehicle type <br> classification | 2.31 s | 4.47 s |

is shown in Table I. For convenience, the ROI is set from the middle to the bottom of each image without considering the upper area because the objects are too small in that area.

The performance of proposed methods is measured by calculating the detection-rate/windows-amount (DR-\#WIN), the vehicle type recognition rate, and the receiver operating characteristic (ROC) curve. The computational time of 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 ground-truth bounding box covered by detected windows is above 0.8 . Vehicle type recognition 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 type 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 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 multi-vehicle by our integrated image cues. In order to describe its ability to extract vehicle proposals, we compute the DR-\#WIN curves of our method on three dataset which is shown in Fig. 6. Different \#WIN represents different candidate location number. A small set of coarse locations with high DR are sufficient for effective vehicle detection, and it allows complex features to be involved in following processing to

TABLE IV
Comparison of Computational Time for Three Methods

| Image Size | Method | Average Processing <br> Time for an Image |
| :---: | :---: | :---: |
|  | Sliding window with CNN | 210 s |
|  | R-CNN | 167 s |
|  | Our method | 5 s |
| $2592 \times 1936$ | Sliding window with CNN | 450 s |
|  | R-CNN | 275 s |
|  | Our method | 12 s |



Fig. 6 DR-\#WIN plots of the triple cue combinations.
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 detection rate for the subsequent vehicle type recognition. This is the reason for efficiency improved in our method. It is crucial to obtain the precise bounding box of each vehicle region before recognition. It also indicates the three cues are complementary and important for finding vehicles in challenging traffic images. And Table III shows detailed information about average processing time in different phases of the proposed method.

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


Fig. 8. Examples for detection and recognition results of test images in each subset. (a) the multi-vehicle type recognition result for a $1920 \times 1080$ image in the daytime,(b) is the multi-vehicle type recognition result for a $1920 \times 1080$ image in the nighttime, (c) the multi-vehicle type recognition result for a $2592 \times 1936$ image in the daytime.

dataset which guarantees the model can be adapted to a variety of situations; second, the robustness depends on multi-feature extractors in different stages. The complementary cues and CNN features also ensure the proposed method can extract features which are invariant to translations, rotations, and noise variances.

Fig. 8 presents examples in the final results of proposed multi-vehicle type recognition method. We sequentially process all the testing images and output the bounding boxes of detected vehicles' type with different color. The red rectangles indicate the bounding boxes of the detected cars, the blue ones indicate the bounding boxes of the detected taxis, and the green ones indicate the bounding boxes of the detected buses. As illustrated in Fig. 8, our method can detect multi-vehicle locations and recognize the corresponding different vehicle type 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. From Fig. 8(c), it can be seen 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 prior objectness measure and CNN-based classifier.

## C. Comparison of Experiments

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

All of the three methods are conducted in GPU mode. The code is implemented in Python, C++ and Matlab. Detailed information of average computational time to process an image for each method is shown in Table IV. It is easy to see our method achieves remarkable advantage in shortening the full computation cost for images in 2 Mega pixels and in 5 Mega pixels. Sliding window fashion and selective search method are time-consuming, requiring hundreds of seconds to process an image. Our method is efficient to decrease the processing time for two main reasons: the first one is integral images are used to efficiently compute three cues for the final objectness score of a window; and the second is a large number of windows have been reduced before the final evaluation by objectness measure in detection. As shown in Table IV, the average processing time for different high-resolution traffic images is no more than 20s in our method. The proposed method is able to efficiently process a $1920 \times 1080$ image with only 5 seconds and a $2592 \times 1936$ image with only 12 seconds 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 detection rate.

In Fig. 9, ROC curves of three methods for multi-vehicle type recognition in subset 1 are shown. The curves are color coded so that the proposed method, R-CNN method and sliding window technique appear as red, green and blue, respectively. Comparison of ROC results clearly shows that our method
achieves remarkable advantages on 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 which couples multi-vehicle detection and classification is more precise in capturing diverse locations of vehicles and classifying their corresponding types. As illustrated in Fig. 9, our method has very strong discriminative power and can achieve the state-of-the-art recognition performance for car, bus, and taxi. Our method has shown its advantage to classify high dimensional features using a single CNN-based model. In addition, it achieves effective performance and is robust to deal 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 method. Three situations cause the most failed cases. Firstly, 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. Secondly, 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. Thirdly, 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 mistaken.

## VI. CONCLUSION

A novel method for multi-vehicle 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 sliding window approach. After that, normalized detection areas are classified into one of three common vehicle types using a single 8 -layer CNN model. Due to the recognition framework, not only vehicle locations are detected, but also vehicle types are determined. Our method has the ability to extract features which 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 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 vehicle recognition process, improving the performance and efficiency of the recognition system. At the same time, we will also expand the network learning dataset 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 University, 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. on Electronics, Communications and Computers, pp. 59-64, 2014.
[5] A. Lu, L. Zhong, L. Li, and Q. Wang, "Moving Vehicle Recognition and Feature Extraction From Tunnel Monitoring Videos," TELKOMNIKA Indonesian J. Electrical Engineering, 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. on Image and Video Processing, 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. Vehicular Electronics and Safety, pp. 110-113, 2013.
[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. Vehicular Electronics and Safety, pp. 105-109, 2013.
[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, 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, 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, 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, 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, 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, 2010.
[16] B. Tian, B. T. Morris, M. Tang, Y. Q. Liu, Y. J. Yao, C. Gou, D. Y. Shen, and S. H. Tang, "Hierarchical and Networked Vehicle Surveillance in ITS: A Survey," IEEE Trans. Intell. Transp. Syst., vol. 16, no. 2, pp. 557-580. 2015.
[17] R. Feris, B. Siddiquie, J. Petterson, Y. Zhai, A. Datta, L. Brown, and S. Pankanti, "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 and Images, pp. 237-244, 2012.
[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, 2012.
[21] I. Endres, and D. Hoiem, "Category independent object proposals," in Proc. European Conf. Comput. Vis., pp. 575-588. 2010.
[22] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. Int. J. Comput. Vis., 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., pp. 1497-1504, 2011.
[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., pp. 580-587, 2014.
[25] Y. Jia, "Caffe: An open source convolutional architecture for fast feature embedding," http://caffe.berkeleyvision.org/, Mar. 2013.
[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, 2012.
[27] Z. Chen, T. Ellis, and S. Velastin, "Vehicle type categorization: A comparison of classification schemes," in Proc. IEEE Conf. Intell. Transp. Syst., pp. 74-79, 2011.
[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.
[30] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in Proc. NIPS, 2014.
[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," arXiv preprint arXiv:1312.6229, 2013.
[32] R. Socher, B. Huval, B. Bath, D. M. Christopher, and Y. N. Andrew, "Convolutional-recursive deep learning for 3d object classification," Advances in Neural Inform. Process. Syst., pp. 665-673, 2012.
[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Inform. Process. Syst., pp. 1097-1105 2012.
[34] P. Sermanet, and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in Proc. IJCNN, pp. 2809-2813, 2011.
[35] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," IEEE Trans. Intell. Transp. Syst., vol. PP, no.99, pp. 1-12, Oct. 2015.
[36] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in Proc. IJCNN, pp. 1453-1460, 2011.
[37] X. Hou, and L. Zhang, "Saliency detection: A spectral residual approach," in Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 1-8, 2007.
[38] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and F.-F., Li, "ImageNet large scale visual recognition competition 2012 (ILSVRC2012)," 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, China.
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 of 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 his Ph.D. in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, New York in 1990.

He joined the University of Arizona in 1990 and became a Professor and Director of the Robotics and Automation Lab (RAL) and Program in Advanced Research for Complex Systems (PARCS). In 1999, he found the Intelligent Control and Systems Engineering Center at the Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Oversea Chinese Talents Program, and in 2002, was appointed as the Director of the CAS Key Lab for Complex Systems and Intelligence Science. From 2006 to 2010, he was Vice President for research, education, and academic exchanges at the Institute of Automation, Chinese Academy of Sciences. Since 2005, he is the Dean of the School of Software Engineering, Xi'an Jiaotong University. 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. His research is focused in social computing and parallel systems, and has published over 10 books and 300 papers in related areas over the past three decades.

Dr. Wang was the Editor-in-Chief of the IEEE Intelligent Systems in 2009 to 2012. He is currently the EiC of the IEEE Transactions on Intelligent Transportation Systems. He has served as General or Program Chair of more than 20 IEEE, INFORMS, ACM, ASME conferences. He was the President of IEEE ITS Society from 2005 to 2007, Chinese Association for Science and Technology (CAST, USA) in 2005, and the American Zhu Kezhen Education Foundation from 2007-2008. Dr. Wang is member of Sigma Xi, an Outstanding Scientist of ACM, and Fellow of IFAC, IEEE, INCOSE, ASME, and AAAS. Currently, he is the Vice President and Secretary General of Chinese Association of Automation. In 2007, he received the 2nd Class National Prize in Natural Sciences of China for his work in intelligent control and social computing. He received IEEE ITS Outstanding Application Award in 2009 and 2015, IEEE ITS Outstanding Research Award in 2011, IEEE Intelligence and Security Informatics Outstanding Research Award in 2012, and 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 SMC Society Norbert Wiener Award.


[^0]:    Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org. This work was supported in part by the National Natural Science Foundation of China under Grant 61304200, 61503380, 71232006, and 61233001 and in part by the Natural Science Foundation of Guangdong Province, China under Grant 2015A030310187.
    Y. Yao, B. Tian, and F.-Y. Wang 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).

