Hybrid Cascade Structure for License Plate Detection in Large Visual Surveillance Scenes

Chunsheng Liu and Faliang Chang

***Abstract*—Though license plate detection has been successfully applied in some commercial products, the detection of small and vague license plates in real applications is still an open problem. In this paper, we propose a novel hybrid cascade structure for fast detecting small and vague license plates in large and complex visual surveillance scenes. For rapid license plate candidate extraction, we propose two cascade detectors, including the *Cascaded Color Space Transformation of Pixel* detector and the *Cascaded Contrast-Color Haar-like* detector; these two cascade detectors can do coarse-to-fine detection in the front and in the middle of the hybrid cascade. In the end of the hybrid cascade, we propose a cascaded convolutional network structure (Cascaded ConvNet), including two detection-ConvNets and a calibration-ConvNet, which is designed to do fine detection. Through experiments with different evaluation data sets with many small and vague plates, we show that the proposed framework is able to rapidly detect license plates with different resolutions and different sizes in large and complex visual surveillance scenes.**

***Index Terms*—License plate detection (LPD), hybrid cascade, convolutional neural networks (CNNs), adaptive boosing (AdaBoost).**

I. Introduction

AUTOMATIC license plate recognition (ALPR) systems have been widely used in automated car parking, traffic surveillance, traffic control, electronic toll collection, vehicle tracking, vehicle localization, and several other applications. As the basic stage of ALPR, license plate detection (LPD) influences the accuracy of ALPR systems. The quality of acquired images is a major factor in the success of an LPD method [1]. With high-quality images, license plate detection and recognition have been successfully applied in some commercial products and are often regarded as a resolved problem. LPD methods also play an important role [2] in some special applications in uncontrolled or complex scenes, such as complex traffic surveillance and long-distance vehicle localization or tracking. Because the license plates captured in these situations are often small, vague and without clear characters, it is difficult to detect license plates in these scenes.

Though lots of LPD methods have been proposed [1], most of these algorithms can achieve excellent results only under certain controlled scenes; in these methods, the license plate characteristics, including boundary, color and texture, are utilized to design detection methods. Yet, for small and vague license plates, the characteristics of boundary, texture and color features are often in inferior quality, which brings difficulties to the detection process.

The cascade structure based on Adaptive boosing (AdaBoost) and Haar-like features has been successfully applied in the LPD problem [3]–[5]. In some applications, the cascaded structure can achieve high detection rate and low false alarm rate. Without directly using remarkable features of boundary, color and texture, the cascaded structure has the potential to detect small and vague license plates. Yet, a trained cascaded detector for license plate detection often cannot achieve good results in detecting small vague license plates, because these license plates usually lack some important features that a well-trained cascade needs. Another shortcoming of a classical cascade structure is that cascading too many similar features may lessen the detection rate.

In recent years, the ConvNet based detection frameworks have achieved high performance in some object detection problems. The ConvNet based methods have been used in the LPD problem [6], [7]. There are also some new nets designed for the small object detection problem, such as the Perceptual GAN net [8] and the Net for finding tiny face [9]; with a process of super-resolution representation or a process of up-sampling image as inputs, these nets are not good solutions for fast object detection in high resolution images. Using the Viola-Jones' cascade structure, the CNN cascade method [10] can partly address this problem. Without exhaustively scanning, the region proposal network (RPN) [11] can predict object bounds and objectness scores at each position and tell the net where to look. The YOLO net [12] can fast detect objects with once scanning; yet, the YOLO network can only deal with fixed input image dimensions and is limited in detecting small license plates [13]. To overcome this problem of YOLO, Silva and Jung [13] trained a frontal-view YOLO-net to detect car frontal-views, and then trained a license plate YOLO-net to detect license plates in the detected frontal-view images; this method is limited in detecting license plates from buses and trucks.

Manuscript received July 7, 2017; revised January 3, 2018 and April 24, 2018; accepted July 8, 2018. Date of publication August 27, 2018; date of current version May 29, 2019. This work was supported in part by the National Nature Science Foundation of China under Grant 61673244 and Grant 61703240, in part by the China Postdoctoral Science Foundation funded Project under Grant 2017M610423, and in part by the Doctor fund of Shandong Province under Grant ZR2017BF004. The Associate Editor for this paper was S. S. Nedevschi. *(Corresponding author: Faliang Chang.)*

The authors are with the School of Control Science and Engineering, Shandong University, Jinan 250061, China (e-mail: liuchunsheng@sdu.edu.cn; flchang@sdu.edu.cn).

Digital Object Identifier 10.1109/TITS.2018.2859348

Based on these analyses, it is found that AdaBoost based cascade structures have the potential to do quick coarse detection, while ConvNet based methods have the potential for fine detection. Combining the advantages of these two structures, we design a hybrid cascaded detector constituted of three cascaded detectors. The process of the proposed method and our main contributions are as follows.

Firstly, based on the observation that license plates usually have distinctive colors, we design a color-pixel based feature called *Color Space Transformation of Pixel* (CST-pixel) feature for fast rejecting backgrounds with other colors. Unlike traditional rectangular features trained by AdaBoost algorithm, the proposed CST-pixel feature is a pixel-level feature. We design threshold-based weak classifiers for the CST-pixel feature, and these weak classifiers can be trained with AdaBoost method into a cascaded CST-pixel detector. The cascaded CST-pixel detector can preserve nearly all license plates and reject backgrounds with other colors. The advantage of the pixel-level cascaded CST-pixel detector is that it does not need the exhaustively scanning process and is robust to ambiguous shapes and textures.

Then, to express contrasted color-region differences in a license plate, we design a new feature called *Contrast-Color Haar-like* (CC-Haar-like) features. More than using only the intensity information, the CC-Haar-like features can utilize the color differences between characters and plate background in a license plate to express local rectangular features. Compared with Haar-like features, the CC-Haar-like features are able to produce more powerful features to express license plates. Furthermore, to avoid over-training of a cascade with only one type of feature, the cascaded CC-Haar-like features detector is trained to be relatively shorter than the classical cascaded Haar-like features and is designed to detect nearly all training plates and reject background samples as much as possible. Though the selected features in a trained cascaded CC-Haar-like features may have sub-optimal ability in rejecting backgrounds, the small and vague license plates are highly tolerated.

Lastly, after the detection process of the cascaded CST-pixel features and the cascaded CC-Haar-like features, there is a relative small number of background subwindows needed to be rejected; in the end of the hybrid cascade, we propose a convolutional network based cascade structure (cascaded ConvNet), including two detection-ConvNets and a calibration-ConvNet, to calibrate detection windows and do fine detection. The detection-ConvNet is trained to detect license plates from the remaining backgrounds. Because the detection windows of the detection-ConvNet may have some deviations and the detection results still have some false alarm samples, the calibration-ConvNet is designed to adjust the detection windows for further background rejection. After the calibration-ConvNet, another detection-ConvNet with more layers is designed to detect license plates from background. Using these three different ConvNets, license plates with different sizes and different clarity can be calibrated and detected.

In this paper, we take the blue Chinese license plates as examples to show our methods. The blue Chinese license plates with different clarity are shown in Fig. 1. Compared with previous methods, the main improvement of the proposed method is that our method has the ability to fast detect license plates with different clarity. As long as the character configurations are given, our method is easily extended to detect different license plates from different countries.

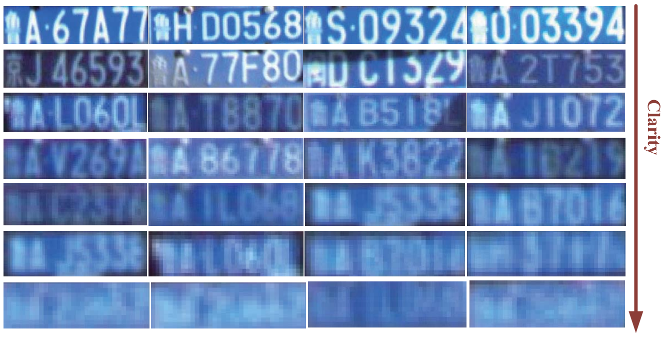


Fig. 1. License plates with different clarity.

The remainder of this paper is organized as follows. Previous license plate detection algorithms are generally classified and reviewed in Section II. In Section III, we present our hybrid cascade for license plate detection in detail. In Section IV, our algorithm is evaluated by the datasets that are collected from real traffic surveillance environments. The experimental results and their comprehensive analysis and discussion are also included in the same section. Finally, we make a conclusion of this paper and present the future work.

II. Related Work

The goal of license plate detection is to locate the regions of interest (ROI) in which a license plate is more likely to be found and verify the hypotheses on the plates presence, result in the candidate of license plates. In the literature, many license plate detection algorithms have been proposed. Although license plate detection has been studied for many years and have many commercial products, it is still a challenging task to detect license plates, especially in inferior quality images. Generally speaking, there are three main methodologies: edge-based methods, color-based methods and texture-based methods [1].

Edge-based approaches are the most popular, with reliable performance in many license plate detection applications. The rectangular shape of the license plate boundary is an important feature that is used to extract license plates. The edge-based detection approaches are mainly based on edge extraction methods, such as Sobel filter [14] and Hough transform (HT) [15]. Since license plate normally has rectangular shape with a known aspect ratio, rectangle detection methods are commonly used in license plate detection [16]–[18]. Some authors prefer detecting vertical edge only because they do not consider the horizontal edge detection to be absolutely reliable due to its similarity of car bumper and other horizontal edges of car [19]–[21]. Without using continuous license plate boundary, blocks with high edge magnitudes are identified as possible license plate areas in [22]. Edge-based methods are often simple and fast when the background is not extremely cluttered and the boundary of license plates are clear and regular. When the edges are not clear or license plates undergo changes from observation, the detection rate would fall rapidly.

Color-based approaches are based on the observation that some countries have specific colors in their license plates. It is intuitive to extract license plates by locating their colors in the images. The color-based detection approaches are mainly based on threshold in some color spaces [23], Genetic algorithm (GA) [24], Gaussian weighted histogram intersection [25], mean shift [26], etc. In [27], the collocation of license plate color and character color is used to generate an edge image. Then, it checks neighbors of pixels with a value within the license plate color range to find candidate license plate regions. To address the effect of illumination variation, [28] proposes a vague logic method for color recognition of license plates. Extracting the license plate using color information has the advantage of detecting inclined and deformed plates. However, it will be very sensitive to various illumination changes and suffer from false positive especially when other parts of testing images have the same license plate color.

Texture-based methods mainly focus on the significant change of pixel intensity between characters and license plate background. Some authors use different image transformation methods for texture analysis, such as Gabor filters [29] and discrete Fourier transform (DFT) [30]. Using different image transformation methods, the texture of different license plates can be enhanced or transformed into some parameters of texture analysis. Blur kernel estimation method [31] can be adopted to reduce the blurring caused by fast motion. Some learning-based methods treat license plate detection as a binary classification problem. [32] applies a support vector machine (SVM) classifier on color texture feature to detect license plates. In [3]–[5], AdaBoost is combined with Haar-like features to obtain cascade classifiers for license plate extraction.

Different from the above three kinds of traditional detection methods, some other approaches based on characters or local features have been proposed. In [33], license plate regions are selected from maximally stable extremal regions (MSER) detection [34] results with some criteria. Without directly extracting license plate regions, some work directly extract and detect characters in license plate. In [35], single character regions are first selected from MSER detection results with some criteria. Then, the conditional random field method is utilized to represent the relationship among the extracted characters in the same license plate.

Though there are so many good methods for license plate detection, no method focuses on the problem of small and vague license plate detection. Distinguished from these previous methods, our method focuses on the small and vague license plate detection problem.

III. Hybrid Cascade for License Plate Detection

A. Motivation and Structure

The AdaBoost and cascade based structure [36] has been proven very efficient for dealing with rare event detection problems because of its asymmetric decision making process; yet, the traditional cascade can only deal with a series of similar rectangle features, such as Haar-like features [36], MB-LBP features [37] or MN-LBP features [38], which largely weakens its ability of detecting complex and changeful objects. Furthermore, when dealing with vague and small license plates, it would be would be very hard to keep a good balance of the detection rate and the false alarm rate by cascading too many features in traditional cascade detectors.

Using Viola and Jones's cascade structure [36], the CNN based cascade method [10] can detect objects in a coarse-to-fine process. Scanning the whole image with different positions and scales, the CNN based cascade method is still very time-consuming when handling high resolution images. If the AdaBoost based cascade method can be utilized to get region proposals for the CNN based cascade method, the detection structure combining the AdaBoost based cascade and the CNN based cascade may be fast and accurate.

Based on the above analysis, we design a hybrid cascade for the detection of license plates with different resolutions. This hybrid cascade structure has three parts including the cascaded CC-Haar-like detector, the cascaded CST-pixel detector, and the cascaded ConvNet detector. The structure is shown in Fig. 2. The principles of this design are as follows.

1) For the traditional AdaBoost based cascade, there are dozens of stages in a cascade and more than one feature in each stage. In training of a cascade, the detection rate is often fixed ranging from 98.0% to 99.5% [36], while the false alarm rate is often fixed around 50.0%; these parameters can ensure that each trained stage can detect 98.0% to 99.5% license plates and reject around 50.0% background subwindows. For LPD, if more stages are cascaded together to achieve lower false alarm rates, the trained cascade detector is often overtrained and misses more license plates, especially the vague and small ones.

To avoid overtraining of a cascade, we use a smaller number of CC-Haar-like features to do coarse detection, with the purpose to retain nearly all plates while rejecting background subwindows as much as possible. With accurate detection rate and relative fast calculation speed, the cascaded CC-Haar-like features are arranged in the middle of the cascade with the purpose of rejecting most backgrounds and detecting nearly all license plates.

2) In the detection process of the cascaded CC-Haar-like features, in order to achieve scale invariance, the input image need to be continuously scaled with scale c into a series of images. c often ranges from $\frac{1}{1.05}$ to $\frac{1}{1.25}$. In vague LPD detection problem, detecting vague plates with small sizes needs a relative large value of c, which is very time-consuming for the scanning process of the cascaded CC-Haar-like features.

Hence, we design the cascaded CST-pixel detector that does not need the scanning process with different scales; arranging the cascaded CST-pixel detector in the front of our hybrid cascade can save a large amount of scanning time.

3) After processing with the front cascaded structures, there are a small number of background subwindows to be rejected. We propose a ConvNet based cascade structure to do fine detection in the end of our hybrid cascade. The proposed

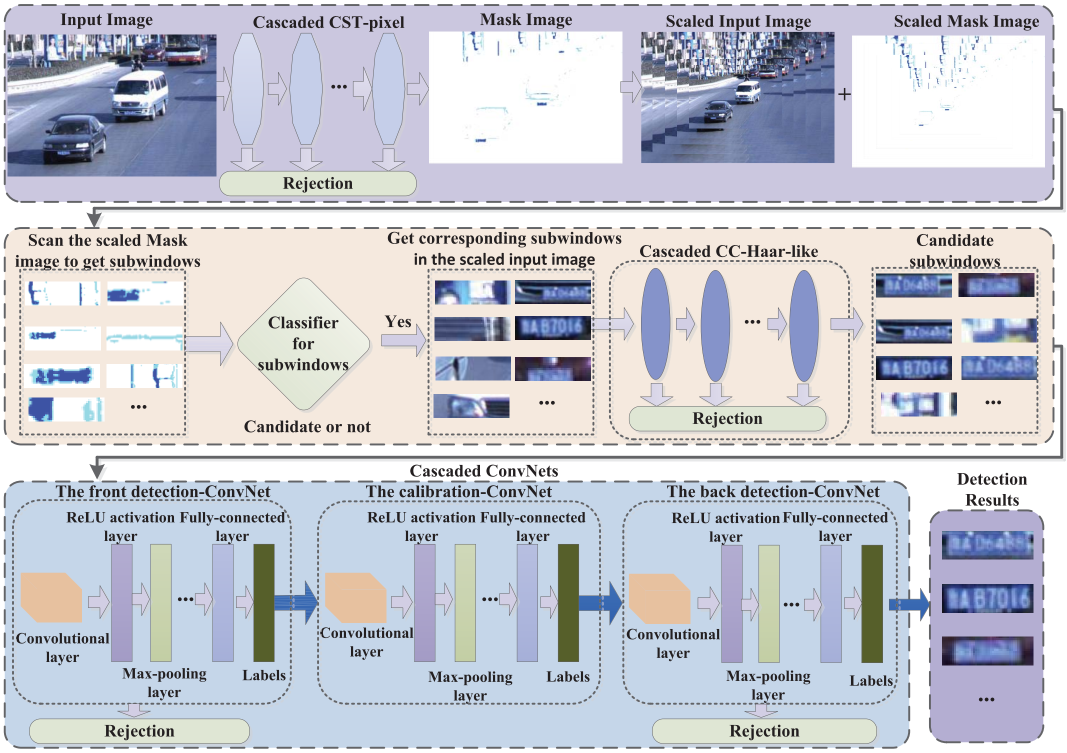


Fig. 2. Hybrid cascade structure for license plate detection.

cascaded ConvNet has two detection-ConvNets and a calibration-ConvNet. The first detection-ConvNet is designed to reject backgrounds. The calibration-ConvNet is designed to calibrate the detection windows for further background rejection. The last detection-ConvNet is designed to do further detection of the windows after calibration. The cascaded ConvNet detector is very time-consuming; arranging this detector in the end of our hybrid cascade can save detection time.

The proposed hybrid cascade detector with three cascaded structures can quickly do coarse-to-fine detection of license plates with different sizes and different clarity. The design of all three parts are described in the following Subsections B, C, and D.

B. Cascaded Color Space Transformation of Pixel Features

Unlike traditional threshold-based color extraction methods, we design a color transformation feature, called *Color Space Transformation of Pixel* (CST-pixel), to transform color information into threshold-based weak classifiers that can be trained with AdaBoost method. Based on the observation that license plates have distinct colors, the color transformation can project the pixel color values in license plates and background into a vector, in which different pixel color values in license plates are clustered and more easily to be extracted with suitable thresholds.

The RGB color space is the most popular color space and can be directly gotten from images and videos. In this study, RGB color space is used for fast calculation, and the channels in RGB space are denoted as r, g, b. The goal of the designed color extraction method is to enhance and extract license plates. We take the detection of blue license plates as an example to show the process. Before training, color training samples for blue license plates and background are established and denoted as $\{(x_1, y_1),\ldots,(x_n, y_n)\}$, where x_i contains three values r_i, g_i, b_i in R, G and B channels, $y_i = \{1, 0\}$ is the training label of color pixels from license plates or background. The framework is shown in Fig. 3 and the training process is shown in Fig. 4. From the structure of the CST-pixel features shown in Fig. 3, it can be seen that the CST-pixel features utilize color space transformation to form weak classifiers of pixels regardless of the input colors. The CST-pixel features are invariant to different color plates.

With different weights, the $x = (r, g, b)$ vector can be weighted and summed into a value p_{ij} in the coordinates (i, j).

$$p_{ij} = (W * x)_{ij} + d, \tag{1}$$

where, W are the convolution weights ranging from -1 to $+1$, and d is the bias.

Then, in the AdaBoost-based learning process for color pixels, the weak classifier of color pixel is defined as:

$$c = \begin{cases} +1, & p_{ij} \cdot \sigma \geq \eta \cdot \sigma \\ -1, & \text{otherwise}, \end{cases} \tag{2}$$

where, η is the threshold to be learned, and $\sigma \in \{+1, -1\}$ is a polarity term, which can be used to invert the inequality relationship between p_{ij} and η.

In the AdaBoost-based learning process, convolution weight W, bias d and threshold η are the parameters to be learned. In the training process, different CST-pixel features

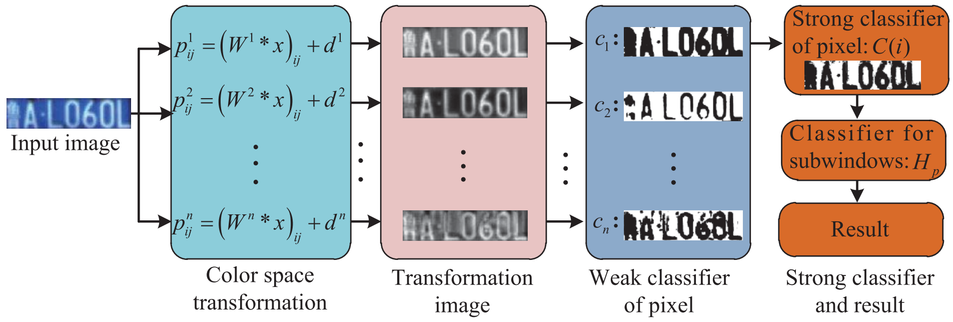

Input image
$p_{ij}^1 = (W^1 * x)_{ij} + d^1$
$p_{ij}^2 = (W^2 * x)_{ij} + d^2$
$p_{ij}^n = (W^n * x)_{ij} + d^n$
Color space transformation
Transformation image
c_1:
c_2:
c_n:
Weak classifier of pixel
Strong classifier of pixel: $C(i)$
Classifier for subwindows: H_p
Result
Strong classifier and result

Fig. 3. Color space transformation of pixel features. The input image utilizes color space transformation to form weak classifiers of pixel; this process is trained by AdaBoost algorithm. Then, the weak classifiers of pixel are selected by AdaBoost algorithm again to form weak classifiers of license plate; the weak classifiers of license plate can form strong classifier to detect license plates.

```
1.(Given)
    $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ : training data.
    $n_p, n_n$ : number of positives and negatives respectively.
    $x_i$ : input pixels.
    $y_i = \{1,0\}$ : class label.
2.(Initialize)
    Initialize weights: $\omega_i^0 = \frac{1}{n_p}, \frac{1}{n_n}$.
3.(Boosting Procedure)
    For $t = 1, \cdots, T$
        (1) Normalize the weights, $\omega_i^{t-1} = \omega_i^{t-1} / \sum_{i=1}^{n} \omega_i^{t-1}$.
        (2) All the CST-pixel features with different convolution
        weight $W$, bias $d$ and threshold $\eta$ are used to select the
        best weak classifier $c_t(x, W, d, \eta)$ with the lowest weighted error:
            $err_t = \sum_{i=1}^{n} \omega_i^{t-1} |c_t(x_i, W, d, \eta) - y_i|$.
        (3) Update the weights: $\omega_i^t = \omega_i^{t-1} \beta_{t-1}^{1-e_i}$,
        where $e_i = 0$ if $x_i$ is classified correctly, $e_i = 1$ otherwise,
        and $\beta_t = err_t/(1 - err_t)$.

    End For
4.(Strong classifier of pixel)
    Strong classifier of pixel: $S = sign(\sum_{t=1}^{T} \alpha_t c_t - \frac{1}{2} \sum_{t=1}^{T} \alpha_t)$,
where, $\alpha_t = log \frac{1}{\beta_t}$.
5.(Cascaded CST-pixel detector)
    Using the training process in [36] to cascade $n_S$ strong classifiers into a cascade detector: $C = \prod_{i=1}^{n_S} S_i$.
6.(Classifier for scanning rectangular subwindows)
    The rectangular subwindow with $n_R$ pixels is denoted as $R$.
    Find a minimum threshold $\theta_S$ to ensure that 99.9% positive
samples satisfying, $\sum_{i=1}^{n_R} C_i \geq \theta_S$.
7.(Output the classifier in scanning:)
    Scanning classifier: $H_p = sign(\sum_{i=1}^{n_R} C_i - \theta_S))$.
```

Fig. 4. Training process of the cascaded CST-pixel detector.

can be selected according to the weighted error function,

$$c_t(x, W, d, \eta) = \arg\min\{\sum_{i=1}^{n} \omega_i^{t-1} |c_t(x_i, W, b, \eta) - y_i|\}, \quad (3)$$

where, x is the input pixel, t denotes the tth iteration step, ω_i^{t-1} is the weight of the ith sample in the $t-1$th interation step, $y_i = \{1, 0\}$ is the class label of the ith sample. The smallest weighted error is $err_t = \sum_{i=1}^{n} \omega_i^{t-1} |c_t(x_i, W, b, \eta) - y_i|$. The weight ω_i^t can be updated as,

$$\omega_i^t = \omega_i^{t-1} \beta_{t-1}^{1-e_i}, \quad (4)$$

where $e_i = 0$ if x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = err_t/(1 - err_t)$.

After T iteration steps, a strong classifier with T weak classifiers can be trained, called color-pixel-classifier. The strong classifier can be calculated as,

$$S = sign(\sum_{t=1}^{T} \alpha_t c_t - \frac{1}{2} \sum_{t=1}^{T} \alpha_t), \quad (5)$$

where, $\alpha_t = log \frac{1}{\beta_t}$; $sign(x) = 1$, if $x \geq 0$ and $sign(x) = 0$, otherwise.

Using the training process in [36] to get a cascade of n_S strong classifiers, called cascaded CST-pixel and denoted as C,

$$C = \prod_{i=1}^{n_S} S_i \quad (6)$$

The image after processing with the cascaded CST-pixel detector is called the mask image M. Because the candidates in the hybrid cascade are rectangles, we need to convert this cascaded detector for pixels into the weak classifiers for rectangle features that can be used in the LPD hybrid cascade.

The input image is scaled into different resolution images for detection. R is denoted as the candidate subwindow in the detection process. n_R is denoted as the pixel number in R, and the sum value of all pixels in R can be calculated as S_R,

$$S_R = \sum_{i=1}^{n_R} C_i, \quad (7)$$

where, C_i is the cascaded CST-pixel of the ith pixel.

Then, in the AdaBoost-based learning and detection process for color pixels in the hybrid cascade, the strong CST-pixel classifier is defined as,

$$H_p = sign(S_R - \theta_S), \quad (8)$$

where, θ_S is the threshold to determine the number of pixels that can be classified as special color pixels.

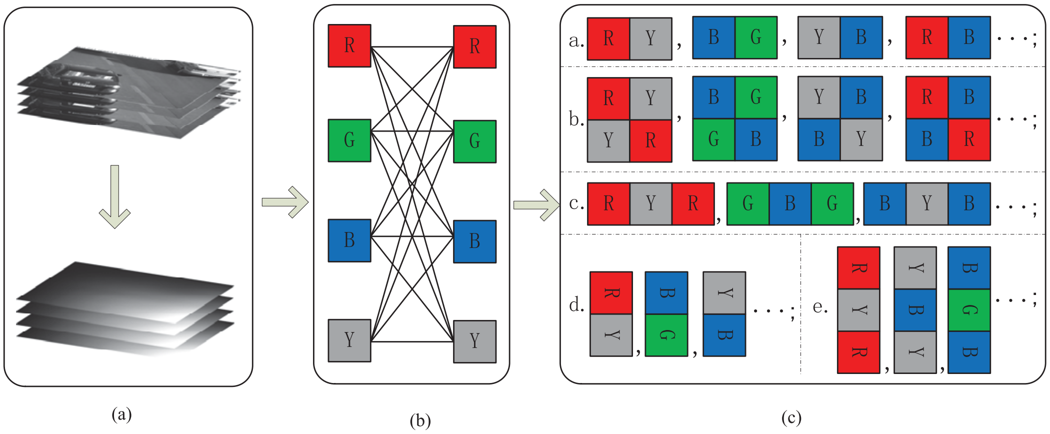


Fig. 5. Contrast-color Haar-like features. In our study, the R, G, B channels in the RGB space and the gray-level channel Y are used to form different CC-Haar-like features. (a) Integral images of different channels. (b) different contrast colors. (c) features of different contrast colors.

The cascaded CST-pixel detector can reject most backgrounds and save detection time for the following detection processes.

C. Cascaded Contrast-Color Haar-Like Features

The most popular features used in boosting-based object detection methods are rectangle features, such as the Haar-like features [36], the MB-LBP features [37] and the MN-LBP features [38]. In the area of license plate detection, the Haar-like features have been successfully applied in some LPD systems; yet, the Haar-like features are limited in expressing small and vague plates and are hard to keep a good balance of detection rate and false alarm rate. Based on the observation that license plates usually have color contrast between characters and background and have similar intervals between adjacent characters, we design a novel feature called *Contrast-Color Haar-like* (CC-Haar-like) feature. Though the MN-LBP features [38] can be utilized to express contrasted colors, the MN-LBP features can only express blue, red and yellow colors using enhanced images and are limited in expressing other colors. The main improvement of the CC-Haar-like features is the introduction of different contrasted color channels, which can largely enhance the ability of expressing contrasted colors in license plates.

The proposed CC-Haar-like features are shown in Fig. 5. To express the differences of contrasted colors, the CC-Haar-like features have different color channels from different color spaces. The gray-level channel and R, G and B channels are used in our experiments; there are two reasons of using these channels: firstly, the CC-Haar-like features calculated with these four channels can express license plates well; secondly, it is time-saving to directly use RGB color space with no other process of color space transformation and calculation. From the structure of the CC-Haar-like features shown in Fig. 5, it can be seen that our CC-Haar-like features can express different contrasted colors in RGB space. The CC-Haar-like features are invariant to the contrasted colors from different color plates.

The integral images for different color channels are calculated,

$$I(x, y, z) = \sum_{x' \leq x, y' \leq y} P(x', y', z), z = 1, 2, ..., N, \tag{9}$$

where $P(x', y', z)$ is the image of the zth channel, and N is the number of channels. The integral image at location (x, y) of the zth channel contains the sum of the pixels above, to the left of (x, y) in the zth channel.

For a CC-Haar-like feature, the sum value r in a rectangle can be calculated as,

$$r = I(x_3, y_3, z) + I(x_0, y_0, z) - I(x_1, y_1, z) - I(x_2, y_2, z), \tag{10}$$

where, $I(x_i, y_i, z)$ is the value of the coordinates (x_i, y_i) in the zth integral image. (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3) are the top-left, top-right, down-left and down-right coordinates of a rectangle respectively.

The CC-Haar-like feature, shown in Fig. 5, is the sum intensity of the rectangle with one color subtracting the sum intensity of the rectangle with another color. The CC-Haar-like features can be denoted as:

$$f = \sum w^{(i)} \cdot r^{(i)} - \sum w^{(j)} \cdot r^{(j)}, \tag{11}$$

where, $\sum w^{(i)} \cdot r^{(i)}$ and $\sum w^{(j)} \cdot r^{(j)}$ are the weighted sum values from i and j color channel respectively; $w^{(i)}$ is the weight assigned to the ith rectangle with default weight $+1$ or -1, and $r^{(i)}$ is the sum intensity of the pixels in the ith rectangle, containing parameters of coordinates of x and y and sizes of m and n.

Then, in the Boost-based learning method, the weak classifier of the CC-Haar-like features is defined as:

$$h_h = \begin{cases} +1, & f \cdot \sigma \geq \theta_h \cdot \sigma \\ -1, & \text{otherwise}, \end{cases} \tag{12}$$

where, θ_h is the threshold to be learned, and $\sigma \in \{+1, -1\}$ is a polarity term, which can be used to invert the inequality relationship between f and θ_h.

The used training method is the AdaBoost learning algorithm [36]; using this training method, we can get the most powerful CC-Haar-like features with suitable parameters. After training, the strong classifier is,

$$H_h = sign(\sum_{j=1}^{T} \alpha_h^j h_h^j - \frac{1}{2}\sum_{j=1}^{T} \alpha_j), \tag{13}$$

where, α_h^j is the weight to be learned like the parameter α_t in formula (5).

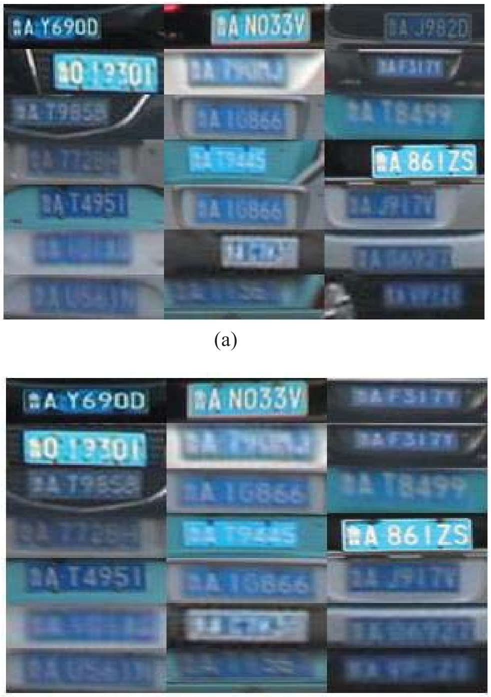
Then the strong classifiers can be connected into a cascade structure using AdaBoost algorithm [36]. In the middle of the hybrid cascade, the cascaded CC-Haar-like features can detect license plates and reject backgrounds. After coarse detection of cascaded CC-Haar-like features, there are a small number of candidates including license plates in good quality or poor quality and some backgrounds. The candidate subwindows can be fast processed with the following cascaded Convolutional Networks.

D. Cascaded Convolutional Networks

The Faster Region-based Convolutional Network methods (Faster RCNN) [11] have achieved excellent object detection accuracy by using a deep ConvNet for detecting object proposals. Some ConvNet based cascade structures are proposed to achieve less detection time [10]. The methods in [10] and [11] need to search objects in a whole image. In our hybrid cascade, there are a number of candidate subwindows processed with the previous two cascades, which can be used as the proposals of the following classifier. Yet, some candidate subwindows processed with the previous two cascades have unpredictable large position deviations (shown in Fig. 6 (a)), which lowers the performance in both detection and localization. To overcome this problem, we propose a convolutional network based cascade structure (cascaded ConvNet) to do fine detection.

The proposed cascaded ConvNet has a front detection-ConvNet, a calibration-ConvNet, and a back detection-ConvNet. The established structure of these three ConvNets is shown in Fig. 7. Trained with the samples with or without large deviations, the front detection-ConvNet is designed to reject part of the background subwindows. Trained with the samples with different types of deviations, the calibration-ConvNet is designed to align the detection windows for further background rejection. Trained with the samples without large deviations, the back detection-ConvNet is designed to do further detection of the subwindows after calibration. The details of the designed cascaded ConvNets are as follows.

After the processing progress of the front two cascade detectors, there are a small number of background subwindows to be rejected. The front detection-ConvNet is trained to detect license plates from the remaining backgrounds. The background samples that cannot be properly classified by the front cascaded structures are added to the negative training set in training the detection-ConvNet. As shown in Fig. 7, the front detection-ConvNet has two repetitions of the three core layers of a 3 × 3 convolution layer, a max-pooling layer and a ReLU activation layer, a 90 × 1 fully connected layer, and a softmax layer.


(a)

(b)

Fig. 6. Part of the candidates before and after calibration. (a) Part of the candidates with unpredictable large position deviations. (b) The calibration results of the images in (a).

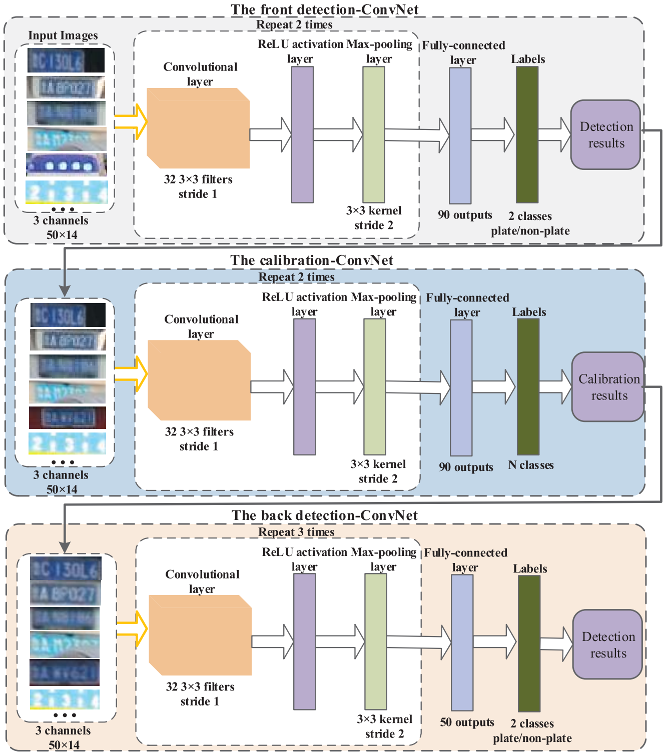


Fig. 7. The Structure of the Cascaded Convolutional Networks.

The detection windows of the front detection-ConvNet may have some deviations, and the detection results still have some false alarm samples. The calibration-ConvNet is designed to adjust detection windows for further background rejection in the next detection-ConvNet. Part of the calibration results are shown in Fig. 6 (b). The calibration-ConvNet has two repetitions of the three core layers of a 3×3 convolution layer, a max-pooling layer and a ReLU activation layer, a 90×1 fully connected layer and a softmax layer. The design of this calibration-ConvNet is similar to the calibration-net design method in paper [10]. N calibration patterns are pre-defined as a set of 3-dimensional scale changes and offset vectors $[s_n, x_n, y_n]_{n=1}^{N}$. Given a detection window (x, y, w, h) with top-left corner at (x, y) of size (w, h), the calibration pattern adjusts the window to be,

$$(x - \frac{x_n w}{s_n}, y, \frac{w}{s_n}, h). \tag{14}$$

After the detection of the previous cascade detector, the detection windows may have deviations of the coordinate x and the width w; the formula (14) is designed to change x and w for calibration. In this work, the calibration-ConvNet has $N = 15$ patterns formed by all combinations of

$$\begin{aligned} s_n &\in \{1, 1.2, 1.4\}, \\ x_n &\in \{-0.2, -0.1, 0, 0.1, 0.2\}. \end{aligned} \tag{15}$$

Given a detection window as the input image, the calibration-ConvNet outputs a vector of confidence scores $[c_1, c_2, ..., c_N]$. The average results of the patterns of high confidence score are taken as the adjustment $[s, x, y]$, i.e.,

$$[s, x, y] = \frac{1}{Z}\sum_{n=1}^{N}[s_n, x_n, y]I(c_n > t), \tag{16}$$

where,

$$Z = \sum_{n=1}^{N} I(c_n > t), \tag{17}$$

and,

$$I(c_n > t) = \begin{cases} 1, & c_n > t \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Here, t is a threshold to filter out low confident patterns. In our experiments, t is a self-adaptive value trained to select the maximum three values of c_n.

After processing with the calibration-ConvNet, the detection windows of the license plate are aligned and more easily classified from backgrounds. The back detection-ConvNet is trained to detect license plates from the remaining backgrounds. The background samples that cannot be rightly classified by the two front ConvNets are added to the negative sample set in training this detection-ConvNet. The last detection-ConvNet has three repetitions of the three core layers of a 3×3 convolution layer, a max-pooling layer and a ReLU activation layer, a 50×1 fully connected layer and a softmax layer. Using these three different ConvNets, license plates with different sizes and different clarity can be aligned and detected.

In the process of selecting the number of layers, some nets with different numbers of layers are trained and tested in our experiments. Because the purpose of the front-ConvNet is to reject part of the background subwindows, we train and select the net with the highest detection rate. Similar to the training process of the front net, we train and select the calibration-ConvNet with the highest IOU (intersection-over-union) and the back-ConvNet with the highest accuracy. In these experiments, we create a set of options for training networks using stochastic gradient descent with momentum. We set the initial learning rate as 0.001 and reduce the learning rate by a factor of 0.1 every 8 epochs; we set the maximum number of epochs for training to 40, and use a mini-batch with 128 observations at each iteration.

IV. Experiments

A. Evaluation Datasets and Setting

Because there is no public comprehensive dataset for evaluating small-size traffic sign detection methods, we establish four testing datasets consisting of Chinese license plate images collected from different video surveillance systems in real road environments at different times during the day. Set 1 contains 142 images with large license plates in local view. Set 2 contains 201 images captured with a surveillance system under large and complex scenes. Set 3 contains 231 images captured with a surveillance system under very large and complex scenes (Scene-A: spring and sunshine). Set 4 contains 228 high resolution images captured with a surveillance system under very large and complex scenes (Scene-B: winter and fog). In the captured images, license plates may appear at different distances from the camera. During our experiments, the trained detector has stable good performance on the plates with more than 12-pixels height. If the height is less than 12-pixels, the license plates become fuzzier and the detection rates fall rapidly. Hence, the license plates with more than 12-pixels height are considered in our experiments. The detailed description of these four datasets are shown in Table I.

The established training set includes 12,590 license plate samples collected from capturing by our lab and downloading from the Internet. For training the cascaded C-CST-pixel detector, we establish a training set containing 96,200 color pixels from these training samples. In training, we did not augment the initial data set to get more training samples. The validation set contains 1,030 images collected from the same sources of the 12,590 extracted license plates. For the AdaBoost based detectors including the cascaded C-CST-pixel detector and the cascaded CC-Haar-like detector, the validation set is used to adjust part of the training samples achieving optimal detectors. For the cascaded ConvNet detector, the validation set is used to choose the optimal net structure. These samples cover different application scenes and can prevent over-fitting. In selecting training samples, we removed the too vague image samples and selected the suitable color pixel samples to prevent under-fitting. In training the cascaded CC-Haar-like detector and the cascaded ConvNet detector, all positive and negative training samples are resized into 50×14. In the training process using AdaBoost algorithm, the goals of the detection rate and the false alarm rate are set as 99.9% and 50%, respectively, with purpose to reject negatives and preserve nearly all positives.

The detection process of the hybrid cascade is similar to classical cascade based detector. To achieve scale-invariant,

TABLE I

LICENSE PLATES COMPOSITION IN THE FOUR TEST DATASETS

Dataset	Details of License Plates in Different Subsets				
	Description	Plate Height	Resolution	Number of Plates	Number of Images
Set-1	Local view	50 ∼ 80 pixels	1616 × 1232	155	201
Set-2	large scenes	12 ∼ 45 pixels	1616 × 1232	235	142
Set-3	Very large scenes (Scene-A: spring and sunshine)	12 ∼ 50 pixels	3648 × 2432	927	231
Set-4	Very large scenes (Scene-B: winter and fog)	12 ∼ 60 pixels	3648 × 2432	1877	228

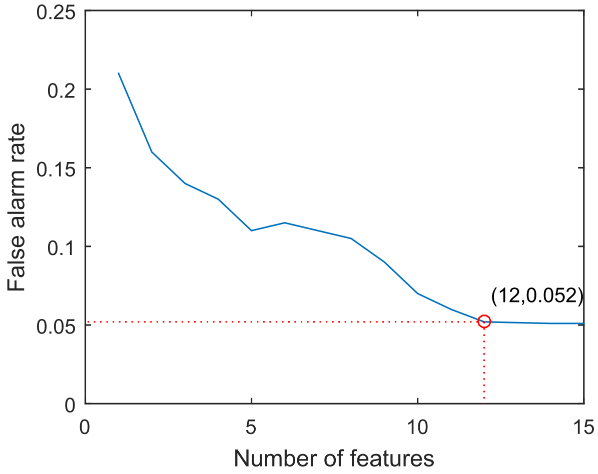


Fig. 8. Curve of the feature number and the false alarm rate in training.

the input image is continuous scaled with scale c into a series of images. c often ranges from $\frac{1}{1.05}$ to $\frac{1}{1.25}$. To detect license plates with small sizes, c is fixed as $\frac{1}{1.05}$ in this study. The 50×14 detector needs to scan the scaled images at a step of three pixels in these experiments. The hybrid cascade detector can scan the image at different scales and different positions resulting in detection results.

The programs are programmed with VC++, OpenCV, Matlab and CUDA on a PC with an Intel i7-7700 CPU, 8GB RAM and an NVIDIA GTX 1060 GPU with 6 GB RAM. The cascaded C-CST-pixel detector and the cascaded CC-Haar-like detector are programmed with VC++ and OpenCV. The cascaded ConvNet detector is programmed based on the RCNN structure provided by Matlab.

B. Evaluation of the Cascaded CST-Pixel Detector

The experiment in this part is designed to demonstrate the cascaded C-CST-pixel can detect regions of interest (ROI) with special colors and reject most backgrounds.

In training the cascaded C-CST-pixel detector, the curve of the feature number and the false alarm rate is shown in Fig. 8. As shown in Fig. 8, the false alarm value tends to lowest when the number of features is twelve, and remains constant when the number of features is bigger than twelve. The C-CST-pixel detector in our experiments has twelve CST-pixel features in three stages.

TABLE II

PERFORMANCE COMPARISON OF DIFFERENT ROI EXTRACTION METHODS

Methods	Set 1, Set 2		Set 3, Set 4		Average	
	DR	ER	DR	ER	DR	ER
RGB-NT [40]	86.3%	97.3%	83.9%	96.8%	84.1%	96.9%
RGB-ET [41]	76.5%	97.7%	73.5%	98.4%	73.8%	98.2%
CMSERs [42]	94.3%	96.6%	91.0%	94.2%	91.3%	94.2%
C-CST-pixel	99.8%	95.0%	99.1%	96.3%	99.2%	95.9%

With purpose to reject backgrounds and detect nearly all license plates, we compare the cascaded CST-pixel features (C-CST-pixel) with three threshold-based color extraction methods including the RGB normalized thresholding (RGB-NT) method [40], the RGB enhancement and thresholding (RGB-ET) method [41], and the Color MSERs (CMSERs) extraction method [42]. Besides the detection rate (DR), we introduce another parameter of the extraction rate (ER) to evaluate the extraction performance. ER is the ratio of the number of extracted pixels to the number of all pixels. The statistical results are shown in Table II.

From the results in Table II, it can be seen that the proposed C-CST-pixel performs much better than the other threshold-based methods in DRs. Though all other methods in comparison can achieve high ERs, the DRs of the RGB-NT and the RGB-ET are just 84.1% and 73.8% on average, which are too low for LPD systems. Because the CMSERs extraction method relies on color enhancement and extreme region detection, the CMSERs method can achieve a relative good DR of 91.3% on average; this DR value is not high enough for vague and small sign detection. The proposed C-CST-pixel method achieves the best DR of 99.2% on average in Table II. The main reasons of this achievement are that the proposed C-CST-pixel can transform the RGB color space into different color spaces that are more suitable for color extraction, and that the cascade structure can combine different threshold-based classifiers into a strong cascaded classifier for color pixel detection. Part of our ROI extraction results are shown in Fig. 9.

C. Evaluation of the Cascaded CC-Haar-Like Detector

The experiment in this part is designed to demonstrate the hypothesis that the cascaded CC-Haar-like features can

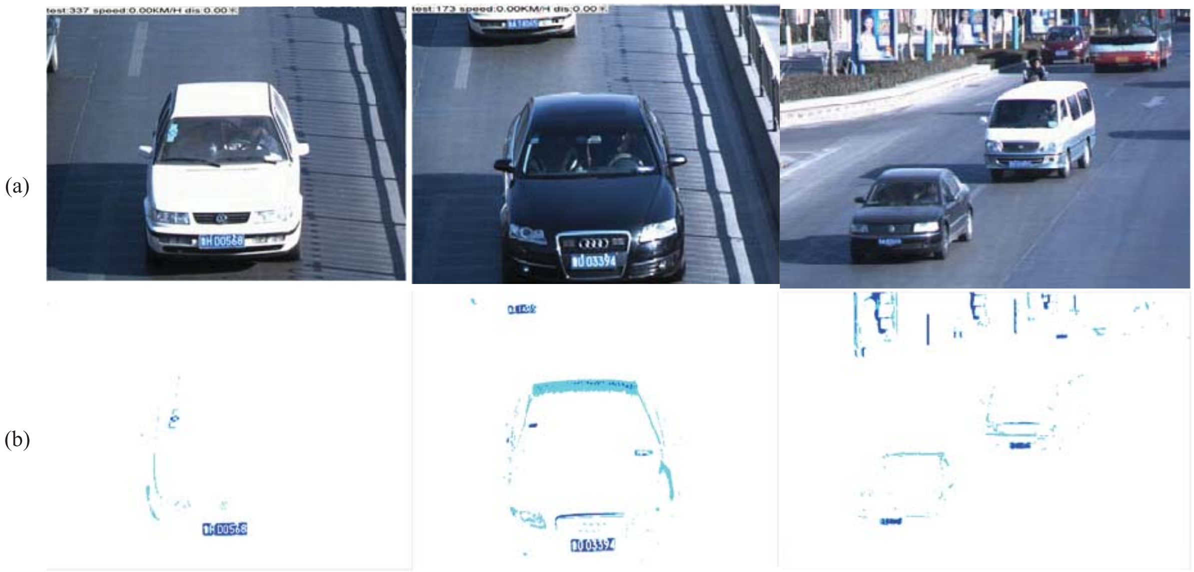

(a)
(b)
test:337 speed:0.00KM/H dis:0.00
test:173 speed:0.00KM/H dis:0.00

Fig. 9. Region of interest extraction results. Images in (a) are the input images. Images in (b) are the ROI extraction results of the input images in (a). The white pixels in (b) mean zero values. The color pixels in (b) are the extracted regions with blue pseudo-color.

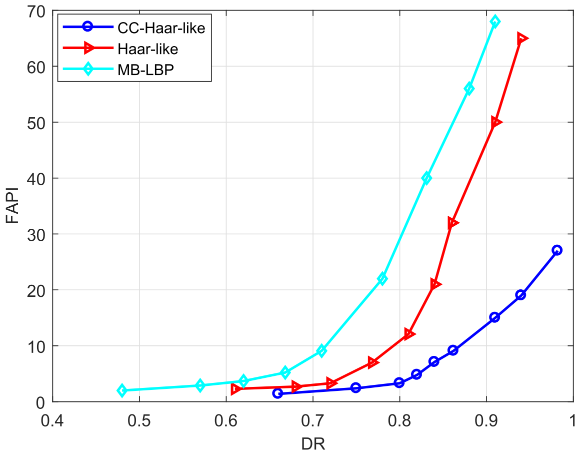


Fig. 10. Curves of the detection rate (DR) and the false alarms per image (FAPI).

effectively detect nearly all license plates in different quality and keep a good balance of detection rates and false alarm rates.

The cascaded CC-Haar-like features are designed for further license plate detection and background rejection. We compare CC-Haar-like features with Haar-like features [36] and MB-LBP features [37]. With the same training set, different cascaded detectors are trained using these three features. The performance is measured by the curves of the detection rate (DR) and the false alarms per image (FAPI), as shown in Fig. 10. All four sets are tested.

From the curves in Fig. 10, it is observed that the cascaded CC-Haar-like features can achieve much less FAPI numbers and a higher DR than the other features. With more than 90% DRs, the other two cascaded detectors have more than 47 false alarms, while the cascaded CC-Haar-like detector has less than 15 false alarms. Achieving the highest DR of 97.1% in our experiment, the cascaded CC-Haar-like features has a FAPI value of 27 on average; the other two cascades have lower highest-DRs of 94.0% and 91.2% respectively, and have much higher FAPIs of 65 and 68 respectively. Such a gain of our method should be mainly attributed to the new structure of CC-Haar-like features, in which the introduction of contrasted colors to classical Haar-like features can produce a large amount of candidate features with strong ability to express license plates. Other traditional gray-level rectangular features in [36] and [37] hardly keep a good balance of DRs and FAPIs.

We test different cascaded CC-Haar-like detectors to get a curve of the feature number and the detection rate (DR), and another curve of the feature number and the false alarms per image (FAPI), shown in Fig. 11. The DRs and FAPIs are the results of different cascaded CC-Haar-like detector with different feature numbers. As shown in Fig. 11, when the cascaded CC-Haar-like detector has 14 stages and 120 features, the hybrid cascade achieves the highest DR of 97.1% and the FAPI of 27. When the feature number is smaller, there are more overlapped detection windows after being processed with the cascaded CC-Haar-like detector, which may result in larger deviations of the detection windows and lower DRs. When the feature number is larger, more license plates are rejected by the cascaded CC-Haar-like detector and result in lower DRs. Selecting suitable number of features can prevent over-fitting or under-fitting. The cascaded CC-Haar-like detector with 14 stages and 120 features are utilized to do detection in our hybrid cascade. All the remaining candidates will be processed using the following cascaded ConvNet detector.

D. Evaluation of the Cascaded Convolutional Networks

As the last detection process in our hybrid cascade, the cascaded ConvNet detector needs to do fine detection of the remaining candidates to achieve high accuracy in both detection and localization. To demonstrate that the designed

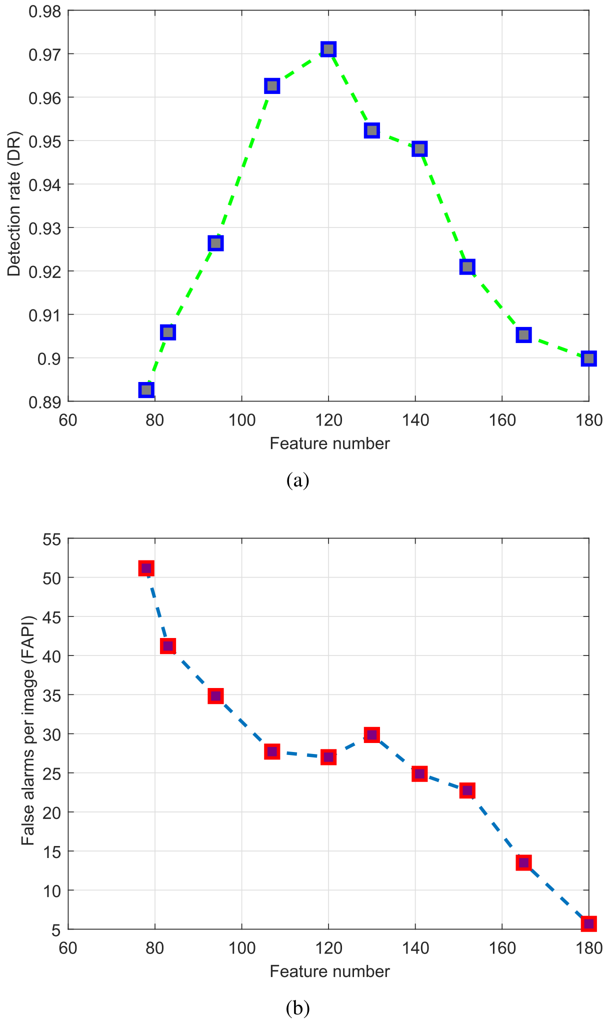


Fig. 11. Performance of the Cascaded CC-Haar-like detector with different feature numbers. (a) Curve of the feature number and the detection rate. (b) Curve of the feature number and the false alarms per image (FAPI).

cascaded ConvNet detector has the ability of detecting small and vague license plates, we compare the cascaded ConvNet detector with three other machine learning based detection methods including the HOG+SVM detector [39], the Haar-like+AdaBoost detector [36] and the RCNN detector [43]. For the HOG+SVM classifier in comparison, a 1944-dimension HOG vector is extracted from each sample, with parameters of cell 5×2, block 10×4, step 5×2 and bin 9. For the Haar-like+AdaBoost classifier in comparison, a cascade detector with 193 Haar-like features is trained. The RCNN detector in comparison has four repetitions of the three core layers of a 3×3 convolution layer, a max-pooling layer and a ReLU activation layer, a fully connected layer and a softmax layer. The statistic data are shown in Table III. The the parameters of DR, FAPI and intersection-over-union (IOU) are used to evaluate the results. The samples that have been processed with the previous two cascade detectors are tested.

The proposed cascaded ConvNet contains three parts including the front detection-ConvNet, the calibration-ConvNet and the back detection-ConvNet. The DR values after the front detection-ConvNet and the back detection-ConvNet are 99.2% and 99.1% on average respectively, which means that the DR value just falls 0.1% after being processed with the calibration-ConvNet and the back detection-ConvNet. The IOU values after the front detection-ConvNet and the calibration-ConvNet are 71.3% and 81.5% respectively, which means that the calibration-ConvNet can effectively reduce 10.2% IOU value. The FAPI numbers after the front detection-ConvNet and the back detection-ConvNet are 0.86 and 0.35 respectively, which means that the calibration-ConvNet and the back back detection-ConvNet can further reduce the FAPI from 0.86 to 0.35.

TABLE III
PERFORMANCE OF DIFFERENT DETECTION METHODS

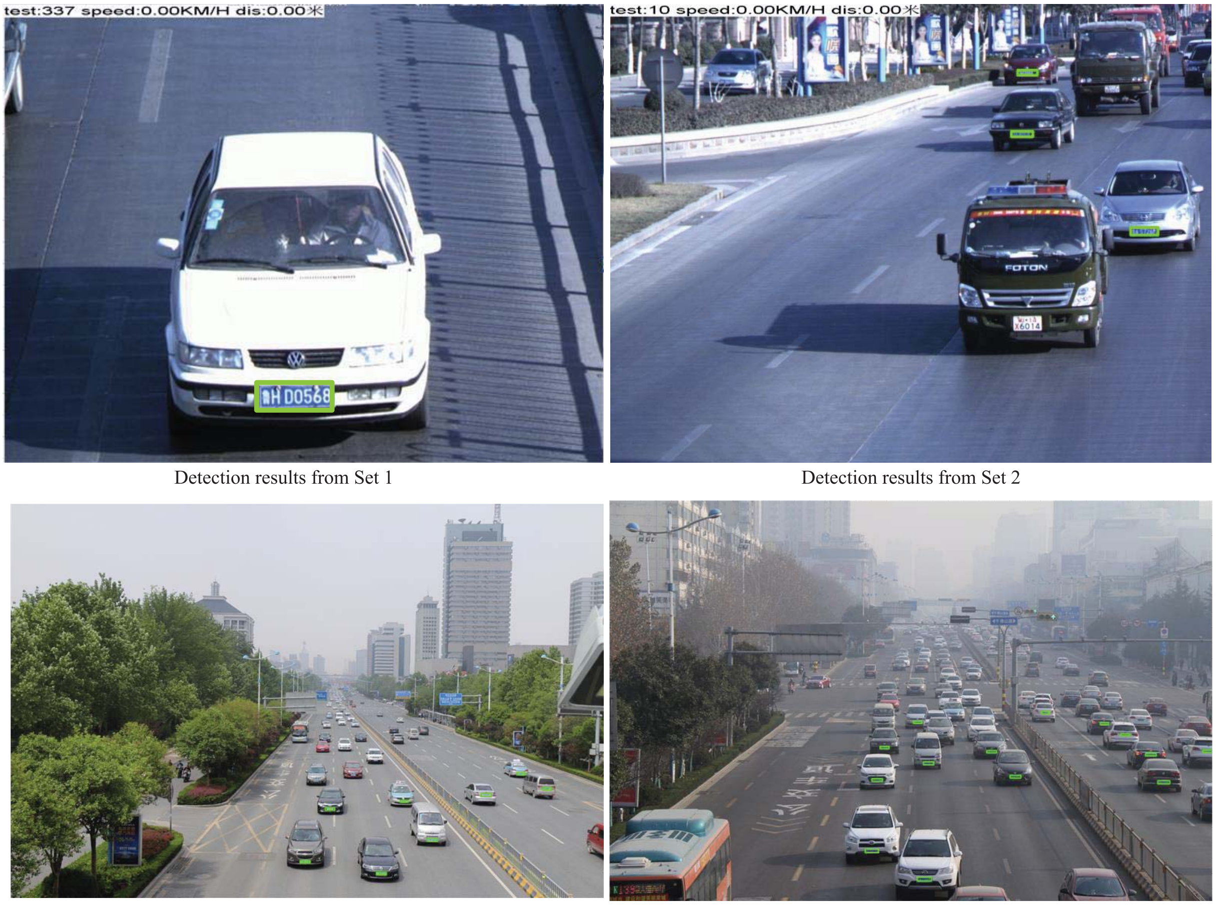
Methods	DR		FAPI	IOU
	Set-1,2	Set-3,4		
HOG+SVM [39]	97.7%	94.8%	2.80	73.3%
Haar-like+AdaBoost [36]	86.0%	84.2%	3.45	77.1%
RCNN [43]	99.8%	98.8%	0.58	72.5%
Cascaded ConvNets	99.8%	99.0%	0.35	81.5%

The statistic results in Table III show that the proposed cascaded ConvNet achieves the highest DRs of 99.8% and 99.0%, the lowest FAPI of 0.35 and the highest IOU of 81.5%. With trials of different HOG features, the chosen HOG features can combine with a linear SVM classifier to achieve relative high DRs in detection. Because the background samples in this comparison are the false alarms processed with the front cascades trained with AdaBoost, the Haar-like+AdaBoost detector has bad performance in classifying license plates from these backgrounds. Achieving similar DRs, our method has a 9.0% higher IOU and a 0.23 lower FAPI than those of the RCNN detector.

There are three main reasons of this achievement. Firstly, with a calibration-ConvNet, the proposed cascaded ConvNet can align the detection results to achieve a high IOU of 81.5%. Secondly, the cascade structure can efficiently reject background subwindows in different cascaded parts achieving a low FAPI of 0.35. Thirdly, the license plates aligned with the calibration-ConvNet can be easier classified with the back-ConvNet achieving better performance in both DR and FAPI. Based on the analysis, it can be concluded that the proposed cascaded ConvNet can achieve high performance in both detection and localization.

E. Performance Evaluation of the Hybrid Cascade Detector

To evaluate the proposed hybrid cascade detector, we use five license plate detection methods for comparison, including the Cascaded Haar-like detector [3], the SVM+HOG detector [39], the OpenALPR detector [44], the CNN based detector [7], the faster-RCNN based detector [11] and the YOLO-Net based detector [13]. The HOG+SVM classifier and the cascaded Haar-like detector have been described in the experimental Subsection D. The OpenALPR detector is a cloud API on the website of OpenALPR [44]. The detailed training processes of both the CNN based detector and the YOLO-net based detector are described in [7] and [13]. Because the described CNN structure in [7] is not designed


test:337 speed:0.00KM/H dis:0.00米
test:10 speed:0.00KM/H dis:0.00米

Detection results from Set 1

Detection results from Set 2

Detection results from Set 3

Detection results from Set 4

Fig. 12. Detection results in different sets.

for small license plates and has large detection windows, we double the input image to address this problem. A faster-RCNN based detector [11] is trained to detect license plates. The method described in [13] trains a frontal-view YOLO-net for car frontal-view detection and a license plate YOLO-net for license plate detection in the detected frontal-view images. Because there are many car rear-views in our set 3 and set 4, the YOLO-net in this comparison is trained for detecting both frontal-views and rear-views. The results are given in Table IV.

The results in Table IV show that our method can achieve the highest average precision of 91.6% and the second highest average recall of 96.4%. For the normal license plate detection in Set 1, all these methods can achieve high performance in both recall and precision. For the license plate detection under large and complex scenes in Set 2, 3 and 4, the traditional methods of cascaded Haar-like detector [3], the HOG+SVM detector[39] and OpenALPR [44] achieve relative low recalls and precisions. The YOLO-net based method [13] needs to detect the frontal-views and rear-views first, and has relative poor performance on bus and rear-view detection; hence, compared with the CNN based method and our method, the YOLO-net achieves a relative low recall of 89.7%. The faster RCNN detector [11] does not have good performance on small-size plate detection, achieving 93.6% reall and 87.3% precision on average. Compared with CNN, our hybrid cascade achieves a 1.2% higher average precision and a 0.8% lower average recall. With enlarged input images and a deep structure, the CNN based method can achieve a better average recall than our method. With different background rejection processes in the hybrid cascade, the average precision of the proposed hybrid cascade is lower than that of the CNN based method.

The hybrid cascade can achieve the highest precision and the second highest recall on average. The main reasons are as follows. Firstly, the cascaded CST-pixel detector can reject backgrounds with color features; without directly using boundary or texture features, the cascaded CST-pixel detector is robust to ambiguous appearance. Then, our method utilizes cascaded CC-Haar-like features to do coarse-to-fine detection; with less number but more powerful color-contrasted features, the cascaded CC-Haar-like detector can highly tolerate vague license plates. Lastly, the cascaded ConvNet detector can accurately detect license plates with few false positives. Hence, it can be concluded that the proposed hybrid cascade LPD method can achieve high recall and precision in the detection of license plates with normal or vague appearances. Part of our detection results are shown in Fig. 12, and part of the extracted plates with different clarity are shown in Fig. 13.

The average detection time of our hybrid is 202.3 ms, which is approximately 1/4300 and 1/2 of that of the

TABLE IV

PERFORMANCE COMPARISON WITH OTHER DETECTION METHODS

Dataset	Hybrid Cascade		Cascaded Haar-like [3]		HOG+SVM [39]		OpenALPR [44]		CNN [7]		Faster-RCNN [11]		YOLO-net [13]	
	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Set-1	**100.0%**	**100.0%**	**100.0%**	95.1%	99.4%	92.8%	99.4%	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
Set-2	94.0%	89.5%	73.2%	31.4%	71.1%	36.1%	26.0%	**100.0%**	96.2%	91.9%	93.6%	86.2%	**97.0%**	90.0%
Set-3	96.3%	88.0%	82.7%	49.5%	84.6%	51.7%	N/A	N/A	**97.0%**	87.3%	94.0%	81.5%	90.3%	**91.0%**
Set-4	96.4%	**93.0%**	79.1%	59.0%	81.0%	56.9%	N/A	N/A	**97.2%**	91.0%	93.2%	90.0%	87.6%	90.9%
Average	96.4%	**91.6%**	80.7%	54.0%	82.2%	54.5%	N/A	N/A	**97.2%**	90.4%	93.6%	87.3%	89.7%	91.3%

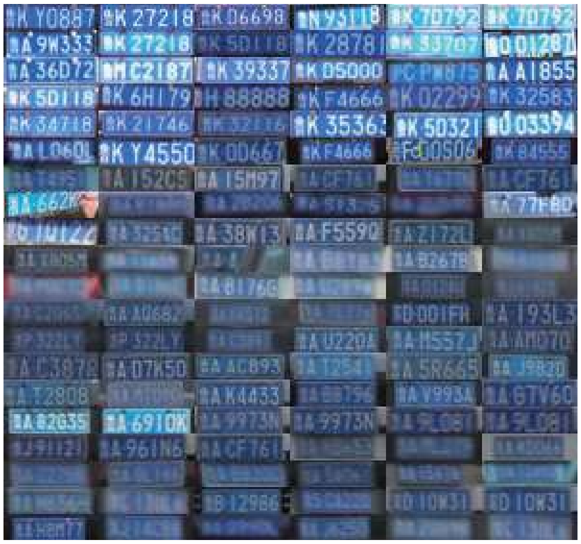


Fig. 13. Detection and extraction results of license plates with different clarity.

CNN-based detector [7] and the YOLO-net based detector [13], respectively. With small-size scanning detection windows and high-resolution input images, the CNN based detector [7] is very time-consuming. Processing the image with one YOLO-net is fast; yet, the YOLO-net based detector [13] has to process each detected car candidate to detect license plates, which largely reduces the detection speed. There are three parts in our hybrid cascade including the cascaded CST-pixel, the cascaded CC-Haar-like and the cascaded ConvNet. The cascaded CST-pixel detector need 9.7 ms on average to reject 95.9% search regions for the following cascaded CC-Haar-like detector. The processing time of the cascaded CC-Haar-like detector is 89.3 ms on average. After processing with the cascaded CC-Haar-like detector, there are only 19 remaining candidates on average to be tested with the following cascaded ConvNet. The cascaded ConvNet needs 103.3 ms to do fine classification. Hence, the proposed hybrid cascade detector can achieve a rapid detection time of 202.3 ms on average. The computing time of our method still has room for improvement. In our cascaded CC-Haar-like part, calculating the integral image of the scaled images is a waste of time. In future, we aim to calculate the integral images of the ROI extraction regions for further improving the running time.

V. CONCLUSION

Several novel methods have been proposed to construct a license plate detection (LPD) system. The presented LPD system can detect and extract license plates with different resolutions and different sizes in large and complex visual surveillance scenes.

To address the problem of fast detecting small and vague license plates, we design a hybrid cascade including three parts: the cascaded CST-pixel detector, the cascaded CC-Haar-like detector, and the cascaded ConvNet detector. Comparing with the traditional cascades based on AdaBoost or CNN, cascading different detectors together can avoid overtraining and detect license plates with different resolutions in high accuracy. The cascaded CST-pixel detector is designed to fast reject backgrounds and is robust to ambiguous shapes. Then, the cascaded CC-Haar-like detector is designed for further background rejection; and in this process the small and vague license plates are highly tolerated. After the detection of the cascaded CST-pixel detector and the cascaded CC-Haar-like detector, there is a relative small number of background subwindows needed to be rejected. The cascaded ConvNet detector is designed to accurately detect license plates. The results of the validation experiments show that the presented hybrid cascade is able to detect license plates with different resolutions and different sizes in large and complex visual surveillance scenes.

This structure can be easily extended to detect license plates with other background colors such as yellow, green, black or white. As a object detection problem, the proposed LPD structure has high potential in other object detection problems such as traffic sign detection and car detection. As a future research work, we aim to improve this structure to apply in other object detection problems in intelligent transportation systems.

REFERENCES

[1] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, “Automatic license plate recognition (ALPR): A state-of-the-art review,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 311–325, Feb. 2013.

[2] B. Tian, Y. Li, B. Li, and D. Wen, “Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 597–606, Apr. 2014.

[3] L. Dlagnekov, “License plate detection using AdaBoost,” Dept. Comput. Sci. Eng., Univ. California, San Diego, La Jolla, CA, USA, Mar. 2004. [Online]. Available: http://www.cse.ucsd.edu/classes/fa04/cse252c/projects/louka.pdf

[4] M. K. Song and M. M. K. Sarker, "Modeling and implementing two-stage AdaBoost for real-time vehicle license plate detection," *J. Appl. Math.*, vol. 2014, Aug. 2014, Art. no. 697658.

[5] H. Zhang, W. Jia, X. He, and Q. Wu, "Learning-based license plate detection using global and local features," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2, Aug. 2006, pp. 1102–1105.

[6] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, "Automatic license plate recognition with convolutional neural networks trained on synthetic data," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Luton, U.K., Oct. 2017, pp. 1–6.

[7] S. Z. Masood, G. Shu, A. Dehghan, and E. G. Ortiz. (2017). "License plate detection and recognition using deeply learned convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1703.07330

[8] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan. (2017). "Perceptual generative adversarial networks for small object detection." [Online]. Available: https://arxiv.org/abs/1706.05274

[9] P. Hu and D. Ramanan. (2017). "Finding tiny faces." [Online]. Available: https://arxiv.org/abs/1612.04402

[10] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proc. CVPR*, Jun. 2014, pp. 5325–5334.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2015.

[12] R. Joseph and F. Ali. (2016). "YOLO9000: Better, faster, stronger." [Online]. Available: https://arxiv.org/abs/1612.08242

[13] S. M. Silva and C. R. Jung, "Real-time Brazilian license plate detection and recognition using deep convolutional neural networks," in *Proc. 30th SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Niteroi, Brazil, Oct. 2017, pp. 55–62.

[14] S. Zhang, M. Zhang, and X. Ye, "Car plate character extraction under complicated environment," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 5, Oct. 2004, pp. 4722–4726.

[15] V. Laxmi, D. K. Mohanta, and B. M. Karan, "Comparison of different wavelets for automatic identification of vehicle license plate," *IET Intell. Transp. Syst.*, vol. 5, no. 4, pp. 231–240, Dec. 2011.

[16] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, and N. Komodakis, "A robust and efficient approach to license plate detection," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1102–1114, Mar. 2017.

[17] C. Gou, K. Wang, Y. Yao, and Z. Li, "Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1096–1107, Apr. 2016.

[18] F. Faradji, A. H. Rezaie, and M. Ziaratban, "A morphological-based license plate location," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Sep. 2007, pp. 57–60.

[19] R. Panahi and I. Gholampour, "Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 4, pp. 767–779, Apr. 2017.

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

[21] S.-Z. Wang and H.-J. Lee, "Detection and recognition of license plate characters with different appearances," in *Proc. IEEE Intell. Transp. Syst.*, vol. 2, Oct. 2003, pp. 979–984.

[22] H.-J. Lee, S.-Y. Chen, and S.-Z. Wang, "Extraction and recognition of license plates of motorcycles and vehicles on highways," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2004, pp. 356–359.

[23] J. Dun, S. Zhang, X. Ye, and Y. Zhang, "Chinese license plate localization in multi-lane with complex background based on concomitant colors," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 3, pp. 51–61, Jul. 2015.

[24] G. A. Samra and F. Khalefah, "Localization of license plate number using dynamic image processing techniques and genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 244–257, Apr. 2014.

[25] W. Jia, H. Zhang, X. He, and Q. Wu, "Gaussian weighted histogram intersection for license plate classification," in *Proc. Int. Conf. Pattern Recognit.*, vol. 3, Aug. 2006, pp. 574–577.

[26] W. Jia, H. Zhang, X. He, and M. Piccardi, "Mean shift for accurate license plate localization," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Sep. 2005, pp. 566–571.

[27] Y.-Q. Yang, J. Bai, R.-L. Tian, and N. Liu, "A vehicle license plate recognition system based on fixed color collocation," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Aug. 2005, pp. 5394–5397.

[28] F. Wang, L. Man, B. Wang, Y. Xiao, W. Pan, and X. Lu, "Fuzzy-based algorithm for color recognition of license plates," *Pattern Recognit. Lett.*, vol. 29, no. 7, pp. 1007–1020, 2008.

[29] H. Caner, H. S. Gecim, and A. Z. Alkar, "Efficient embedded neural-network-based license plate recognition system," *IEEE Trans. Veh. Technol.*, vol. 57, no. 5, pp. 2675–2683, Sep. 2008.

[30] R. Parisi, E. D. Di Claudio, G. Lucarelli, and G. Orlandi, "Car plate recognition by neural networks and image processing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 3, May 1998, pp. 195–198.

[31] Q. Lu, W. Zhou, L. Fang, and H. Li, "Robust blur kernel estimation for license plate images from fast moving vehicles," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2311–2323, May 2016.

[32] K. I. Kim, K. Jung, and J. H. Kim, "Color texture-based object detection: An application to license plate localization," in *Proc. Int. Workshop Pattern Recognit. Support Vector Mach.*, Aug. 2002, pp. 293–309.

[33] V. Seetharaman, A. Sathyakhala, N. L. S. Vidhya, and P. Sunder, "License plate recognition system using hybrid neural networks," in *Proc. IEEE Annu. Meeting Fuzzy Inf.*, vol. 1, Jun. 2004, pp. 363–366.

[34] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.

[35] B. Li, B. Tian, Y. Li, and D. Wen, "Component-based license plate detection using conditional random field model," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1690–1699, Dec. 2013.

[36] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[37] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face detection based on multi-block LBP representation," in *Proc. Int. Conf. Biometrics*, 2007, pp. 11–18.

[38] C. Liu, F. Chang, Z. Chen, and D. Liu, "Fast traffic sign recognition via high-contrast region extraction and extended sparse representation," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 79–92, Jan. 2016.

[39] R. F. Prates, G. Cámara-Chávez, W. R. Schwartz, and D. Menotti, "Brazilian license plate detection using histogram of oriented gradients and sliding windows," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 6, pp. 39–52, 2013.

[40] H. Gómez-Moreno, S. Maldonado-Bascón, P. Gil-Jiménez, and S. Lafuente-Arroyo, "Goal evaluation of segmentation algorithms for traffic sign recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 917–930, Dec. 2010.

[41] A. Ruta, Y. Li, and X. Liu, "Real-time traffic sign recognition from video by class-specific discriminative features," *Pattern Recognit.*, vol. 43, no. 1, pp. 416–430, 2010.

[42] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1498–1506, Dec. 2012.

[43] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, Jun. 2014, pp. 580–587.

[44] OpenALPR. *OpenALPR Cloud API*. Accessed: Feb. 2018. [Online]. Available: http://www.openalpr.com

Chunsheng Liu received the B.S. degree in industrial automatic, the M.S. degree in pattern recognition and machine intelligence, and the Ph.D. degree in pattern recognition and machine intelligence from Shandong University, Jinan, China, in 2009, 2012, and 2016, respectively. He is currently a Research Assistant at the School of Control Science and Engineering, Shandong University. His research interests include pattern recognition, machine learning, intelligent transportation system, and object detection.

Faliang Chang received the B.S. and M.S. degrees from Shandong Polytechnic University, Jinan, China, in 1986 and 1989, respectively, and the Ph.D. degree in pattern recognition and intelligence systems from Shandong University, in 2003. Since 2003, he has been a Professor in pattern recognition and machine intelligence with the School of Control Science and Engineering, Shandong University. His current research interests include computer vision, image processing, intelligent transportation system, and multi-camera tracking methodology.