License Plate Localization in Unconstrained Scenes Using a Two-Stage CNN-RNN

Jingjing Zhang, Yuanyuan Li, Teng Li, Lina Xun, and Caifeng Shan

***Abstract*— Recent deep object detection methods neglect the intrinsic properties of the license plate, which limits the detection performance in unconstrained scenes. In this paper, we propose a two-stage deep learning-based method to locate license plates in unconstrained scenes, especially for special license plates such as fouling, occlusion, and so on. A deep network consisting of convolutional neural network (CNN) and recurrent neural network is designed. In the first stage, fine-scale proposals are detected according to the characteristics of the license plate characters, and CNN is used to extract the local features of characters. A vertical anchor mechanism is designed to jointly predict the position and confidence of each fix-width character. Furthermore, the sequential contexts of characters are modeled with the bi-directional long short-term memory, which greatly improves the locating rate of license plates in complex scenes. In the second stage, the whole license plate is obtained by connecting the fine-scale proposals. The experimental results show that the proposed method not only locates license plates of different countries accurately but also be robust to scenes of illumination variation, noise distortion, and blurry effects. The average precision reaches 97.11% on multi-country license plates, and the precision and recall reaches 99.10% and 98.68%, respectively, on Chinese license plate images.**

***Index Terms*— License plate localization, convolutional neural networks, recurrent neural networks, BLSTM.**

I. Introduction

LICENSE plate recognition (LPR) has a wide range of applications such as urban security, parking management and security control. Typical LPR framework has a license plate localization followed by an optical character recognition. The goal of license plate localization is to accurately predict license plate's bounding box. However, most existing algorithms work well only under controlled conditions or with complex image capture systems, such as viewpoint, specific license plate color [1], fixed size [2] or license plates of only one country [3], [4]. Accurate localization of license plates in an uncontrolled environment remains a challenging task. The difficulty lies in highly complex backgrounds, windows, guardrails or bricks, as well as random shooting conditions such as lighting, distortion, occlusion or blurring.

Fig. 1. License plate examples: left images show obscured ones while right shows defaced ones.

State-of-the-art methods for unconstrained object detection are usually based on Faster R-CNN framework [5], where a Region Proposal Network (RPN) is proposed to generate high-quality class-agnostic object proposals directly from convolutional feature maps. Then an effective anchor regression mechanism is proposed which can detect multi-size objects using a single-size sliding window. The key insight is that a single window can predict objects of various scales and aspect ratios by using multiple flexible anchors. However, Faster R-CNN based object detection techniques aim to handle general object detection problem, license plate localization has unique difficulties we often encounter some of the defaced and obscured license plates, as shown in Fig. 1. Due to obstructing and fouling parts, which breaks the region of a license plate, it is very difficult for a general object detection network to train a model that locates this license plate accurately.

The license plate has two intrinsic properties that could facilitate detection: first, license plate are constituted by fix-length characters, even from different countries with different sizes, character fonts, colors, and distortions; second, the license plate characters are arranged sequentially and horizontally, each occupying a fixed proportion. To take advantage of the high semantic feature representation capacity of deep learning, and handle different patterns and model context properties of license plate simultaneously, we introduce a two-stage unconstrained license plate localization method based on CNN-RNN. The network uses LSTM to extract contextual information and combines with CNN to achieve good results. Similarly in [6] and [7], the local word feature is represented by CNN and the sequential contexts are modeled with BLSTM. But their main task is Named Entity Recognition in text processing area, and the input is word embeddings and character embeddings, while our input is the entire picture. In addition, since the feature representations for character embeddings are much less than those for images, the light

Manuscript received November 20, 2018; revised February 11, 2019; accepted February 12, 2019. Date of publication February 19, 2019; date of current version June 4, 2019. This work was supported in part by the Anhui Provincial Natural Science Foundation under Grant 1608085MF136 and Grant 1808085MF209, in part by the National Science Foundation for China under Grant 61602002 and Grant 61572029, in part by the Open Fund for Discipline Construction, Institute of Physical Science and Information Technology, Anhui University, and in part by the Scientific Research Development Foundation of Hefei University under Grant 19ZR15ZDA. The associate editor coordinating the review of this paper and approving it for publication was Prof. Kazuaki Sawada. *(Jingjing Zhang and Yuanyuan Li contributed equally to this work.) (Corresponding author: Teng Li.)*

J. Zhang, Y. Li, T. Li, L. Xun are with the College of Electrical Engineering and Automation, Anhui University, Hefei 230601, China (e-mail: liteng@ahu.edu.cn).

C. Shan is with Philips Research, High-Tech Campus, 5656AE Eindhoven, The Netherlands.

Digital Object Identifier 10.1109/JSEN.2019.2900257

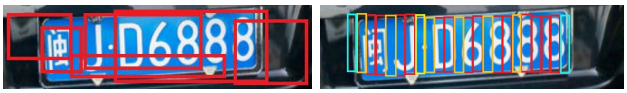

Fig. 2. Left image shows RPN proposals while right image shows fine-scale character proposals.

network model they used is not suitable for license plate localization based on image processing.

In the first stage of our method, we apply an anchor mechanism to locate license plate by characters. Generic object detection methods usually assume that each object has a well-defined closed boundary [8], which may not be suitable for characters consisting of many strokes. Moreover, finding a clear boundary would be quite difficult in low-quality pictures. To address this issue, we introduce fine-scale proposals, where each proposal represents a part of characters. As shown in Fig. 2 (left), conventional RPN [5] does not precisely locate characters. So we develop the vertical anchor mechanism that predicts the confidence of character and y-axis positions of each fine-scale proposal. Fig. 2 (right) shows the result that a sequence of fine-scale (e.g., fixed 16-pixel width) character proposals can locate most characters, and we can locate the license plate accurately by connecting fine-scale proposals. Moreover, the fixed width property can reduce the search space and computation compared to RPN.

In the second stage, our method combines the fine-scale proposals obtained by CNN-RNN, where every fine-scale proposal has a confidence score and we connect the fine-scale proposals to obtain the region of the license plates according to the confidence score.

The main contributions of this work are summarized as follows:

- We deem the license plate localization as a sequence of fine-scale character proposals generation and connection problem, which largely improves the precision.
- We propose the framework combing CNN and BLSTM, which can model both local and global contextual information.
- Extensive experimental results show that our proposed method can not only handle license plates of different countries but also be robust to scale or lighting variation, as well as defaced, fuzzy, obscured and low-quality imaging conditions. Therefore, the proposed method can be applied in practical scenarios with difficult conditions such as hard weather, poor lighting, etc. It also can be widely used in handheld devices, other applications on highway, in crowded driveway, and so on.

The rest of the paper is organized as follows. Section II gives a brief discussion on related work. Section III introduces the details of the proposed method. In Section IV, we report and analyze extensive experimental results. Finally, we conclude the paper in Section V.

II. Related Work

Conventional handcrafted feature based methods target at samples from controlled conditions or relatively single environments. In [10], the proposed plate extraction method consists of edge detection, edge image binarization via adaptive thresholding (AT) and a novel line density filter. In [11], the boundary map is obtained by the Canny edge detector and the unwanted horizontal background edges are removed. The boundaries are then classified into different clusters by density-based methods and then support vector machines is adopted to detect license plates. Reference [12] proposed to convert the license plate image to HSV color space, and detect license plates through cascaded adaboost and multi-block local binary pattern (MB LBP). Color-based or other low-level feature based methods are computationally efficient, and they achieve successes in constrained scenes such as parking place entrance. However, they are too sensitive to illumination change, various viewing angle, stains, occlusion, and image blur in license plate images in natural scenes.

Recently, deep learning based methods have shown impressive performances in various vision tasks [13]–[16], such as face detection, object localization, image classification, etc. Using CNN to extract the semantic features from the image region [17] greatly improves the performance of license plate localization. In [18], they proposed You Only Look Once (YOLO) and fine tuning achieve a good detection and recognition effect on the Brazilian license plates data set. The proposed method requires extracting the Frontal-View before detecting the license plate, which limits its applicable scenarios. In [19], the proposed method uses successive mean quantization transform (SMQT) to extract feature information at few scales, and then use a fixed sliding window to extract candidate boxes. Reference [20] proposed to apply the CNN network to extract features from candidate regions based on the sliding window, and finally obtain the license plate regions by the support vector machine. However, fixed sliding window will cause some final extracted candidate boxes to be too large or too small.

In this paper, we focus on license plate localization in unconstrained scenarios, such as arbitrary surveillance video, hand captured images, and etc. Our proposed method performs well for occlusion, fouling, blurred image, uniform illumination, and even multi-country license plates localization.

III. License Plate Localization

A. Fine-Scale Character Proposals Generation

The fine-scale character proposals generation process is illustrated in Fig. 3.

The CNN model we designed includes five convolution layers extracts the license plate features. Excessive convolution layers will lead to missing detection of possible license plates due to the reduced number of anchors. Too few convolutional layers will lead to false-alarm detections due to the increased number of anchors. It has been verified through experiments that using five convolution layers can achieve good results. The detailed model structure is shown in Table I. The input image of the CNN model is a normalized image. The first convolution layer consists of 64 feature maps, each neuron on the feature map is connected to a 3×3 filter on the input image. The same 3×3 connection weight is shared in each feature map. In order to reduce the number of model parameters and

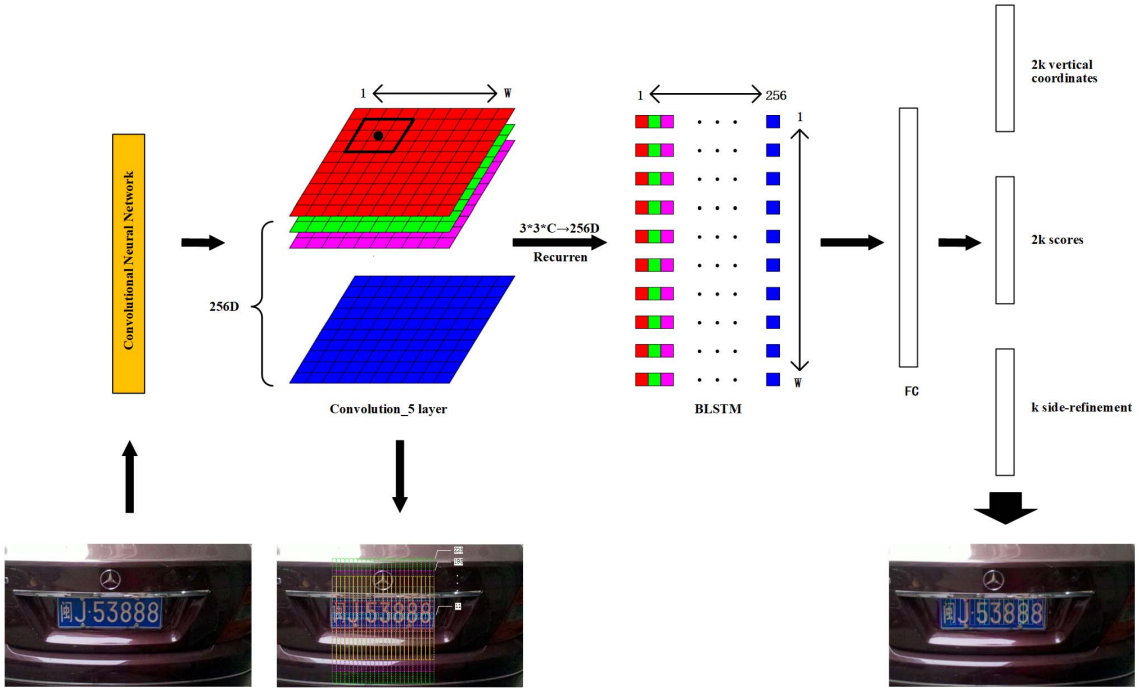

Fig. 3. The first stage of the proposed method: we densely slide a 3 × 3 spatial window on the convolution_5 layer. The sequential windows in each row are recurrently connected by a Bi-directional LSTM (BLSTM) [9], where the convolutional feature (3 × 3 × C) of each window is used as input of the 256D (D represent dimension) BLSTM. The BLSTM layer is connected to a 512D fully-connected layer, followed by the output layer, which jointly predicts confidence scores, y-axis coordinates and boundary-refinement offsets of k anchors.

TABLE I
CONFIGURATION OF THE CNN

Layer Type	Parameters
Convolution_5	#filters:256, k:3×3, s:1, p:1
Maxpooling	p:2×2, s:2
ReLU	
Convolution_4	#filters:256, k:3×3, s:1, p:1
ReLU	
Convolution_3	#filters:256, k:3×3, s:1, p:1
Maxpooling	p:2×2, s:2
ReLU	
Convolution_2	#filters:256, k:3×3, s:1, p:1
ReLU	
Convolution_1	#filters:256, k:3×3, s:1, p:1
Input	color image

avoid the model over-fitting problem, the sub-sampling layer is connected to the convolution layer. After convolution_5, we will get 256 feature maps that contain most of the image features we want.

We obtain the fine-scale proposals on the original image by convolution_5 layer. We use the 3 × 3 space window to slide intensively on the convolution_5 feature maps. It locates a series of characters by sliding a small window on the convolution feature map, and outputs a series of fine character proposals, as shown in Fig. 3. Convolution_5 feature maps size is determined by the size of the input image, and the total step fixed at 16.

We designed the fine-scale character proposals generation method as follows: Our detector densely investigates each space position in convolution_5 layer. The character proposal is defined as a fixed width (in the input image) with 16 pixels. That means we let the detector move densely through the convolution_5 feature map, where the total stride is exactly 16 pixels. Then we will design k vertical anchors at the same horizontal position to predict each proposal. The k vertical anchors have fixed width but have different vertical coordinates. In our experiments, we set the value of the k as 10 and the heights of the anchors vary from 11 to 283 pixels (each/0.7). The explicit vertical coordinates are measured by the height of the proposed bounding box and the y-axis center. We calculate the predicted vertical coordinates relative to the position of the bounding box of the anchor as

$$y_c = (m_v - m_v^a)/h^a, \quad y_h = \log(h/h^a) \tag{1}$$

$$y_c^* = (m_v^* - m_v^a)/h^a, \quad y_h^* = \log(h^*/h^a) \tag{2}$$

where $\boldsymbol{y} = \{y_c, y_h\}$ and $\boldsymbol{y}^* = \{y_c^*, y_h^*\}$ are the relative predicted coordinates and ground truth coordinates, respectively. m_v^a and h^a respectively are the midpoint (y-axis) and height of the anchor box, which can be calculated from the original image. m_v and h are the middle and height predicted coordinates on the y-axis of the input image, respectively, while m_v^* and h^* are the ground truth y-axis coordinates. And the width of the bounding box for each predicted character proposal is fixed, so the shape of the bounding box as shown in Fig. 2 (right). In the figure, the color of each box indicates the confidence score. In order to distinguish different confidences

Fig. 4. The two images show that model without BLSTM will generate negative proposals of similar characters.

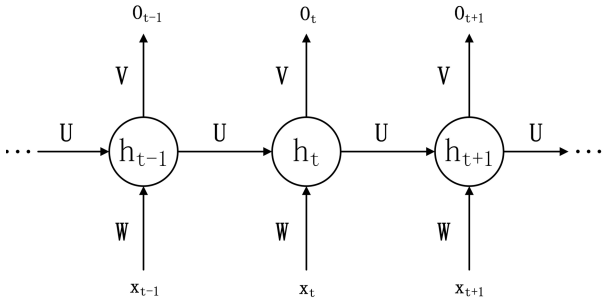

Fig. 5. Structure of a basic recurrent layer.

more clearly, we set the colors of red, orange, yellow, green, cyan, blue, and violet to indicate the confidence scores from high to low. Only the boxes with positive scores are presented.

Given the input image, we get a feature map of W $\times$ H $\times$ C through five convolution layers, where C represents the number of feature maps and W $\times$ H is the spatial arrangement. When our detector densely slides on the convolution_5 layer with a 3×3 window, each sliding window takes a $3 \times 3 \times$ C convolution feature to produce a prediction. For each prediction, horizontal positioning and k-anchor positioning are fixed, which can be pre-calculated by mapping the spatial window position in the convolution_5 layer on the input image. At each window location, our detector outputs the confidence scores and the predicted y-coordinates (y) of k anchors. The confidence score of the detected character proposals are generated from the anchors is greater than 0.7 (with non-maximum suppression). If the value of confident score is set too high, many text anchors will be missed, which will reduce the detection accuracy of the license plates such as insults, obstructions, blurs and so on. However, if the value of confident score is set too low, we will retain many non-text anchors, which will lead to the detection of many non-license bounding boxes. Therefore, the experimental verification is used to set the confident score as 0.7, that can achieve good results.

B. CNN-BLSTM

The localization with a series of fine-scale characters proposals could fail in the existence of non-character objects (such as windows, bricks, leaves, etc.) with a similar structure of character patterns, as shown in Fig. 4. The characters have powerful sequential features, and sequential contextual information plays an import role in image segmentation and recognition tasks, which is confirmed by a recent work [21] and the RNN is used to encode the context information for the text recognition. Inspired by [21], we believe that continuous contextual information may also be important for our detection tasks. Our detectors can use these important contextual information to make more reliable decisions, when it works on each individual work proposal.

Our goal is to re-encode spatially sequential contexts in the convolution layer, and result in a graceful seamless network connection of fine-scale characters proposals in the license plate. RNN provides a natural choice that can recursively encode this information with its hidden layer. The structure of a basic RNN with one recurrent layer is illustrated in Fig. 5. Here the convolution features of windows on the 5-th layer $x_1, x_2, \ldots, x_T$ are used as sequential input to the RNN, and $h_1, h_2, \ldots, h_T$ is a sequence of hidden sates.

A recurrent layer in RNN has a recursive function f, which takes as input the vector x_t and the previous hidden state h_t and outputs new hidden state as:

$$h_t = f(x_t, h_{(t-1)}) = tanh(Wx_t + Uh_{(t-1)}), \tag{3}$$

where W, U and V are the weight matrices, which are shared across all steps, and activation function tanh represents the hyperbolic tangent function.

In practice, RNN often encounters gradient explosions in training, and LSTM can solve this problem by introducing three additional multiplication gates [22]. The internal state in the RNN hidden layer can model the sequential context information that all previous windows scanned. We further extend the RNN layer by using BLSTM, which can encode the recurrence contexts in two directions, so that the connector receive field can cover the entire image. The LSTM unit consists of four sub-units as shown in Fig. 6: input gate, output gate, forget gate and new memory, which are computed by:

$$g^{(t)} = \phi(W_{gx}x^{(t)} + W_{ih}x^{(t-1)} + b_g) \tag{4}$$
$$i^{(t)} = \sigma(W_{ix}x^{(t)} + W_{ih}x^{(t-1)} + b_i) \tag{5}$$
$$f^{(t)} = \sigma(W_{fx}x^{(t)} + W_{fh}x^{(t-1)} + b_f) \tag{6}$$
$$o^{(t)} = \sigma(W_{ox}x^{(t)} + W_{oh}x^{(t-1)} + b_o) \tag{7}$$
$$s^{(t)} = g^{(t)} \odot i^{(t)} + s^{(t-1)} \odot f^{(t)} \tag{8}$$
$$h^{(t)} = s^{(t)} \odot o^{(t)} \tag{9}$$

where activation functions ϕ and σ are logistic sigmoid and hyperbolic tangent functions, respectively. $\odot$ denotes the point-wise multiplication of two vectors.

In model training the internal state in the hidden layer is updated recursively, H_t

$$H_t = \varphi(H_{t-1}, X_t), \quad t = 1, 2, 3, \ldots, W \tag{10}$$

where $X \in R^{3\times3\times C}$ is the input feature of t-th sliding-window (3×3) from the 5-th convolution layer. The sliding window moves densely from left to right, which produces t = $1, 2, \ldots, W$ sequential features for each row. The hidden states of H_t are a sequence of high-level features from the current

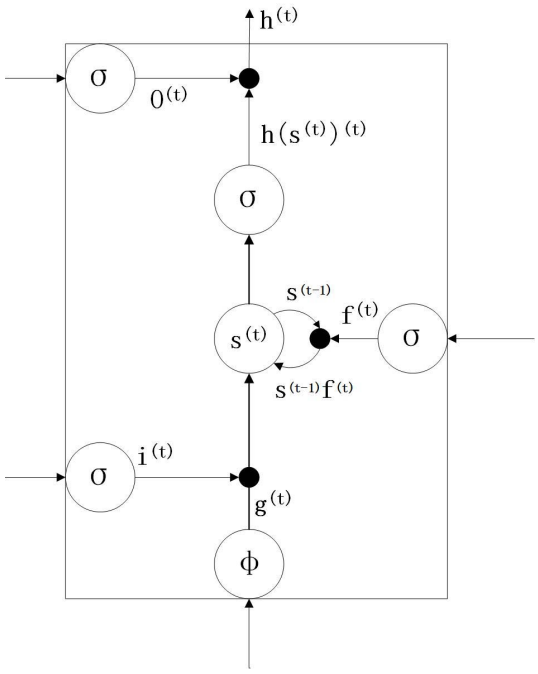

Fig. 6. Structure of a basic LSTM layer.

input X_t and previous state encoded in $(H_{t-1}, H_{t-2}, \ldots)$. We only take the last hidden state of the last recurrent layer as the output, since the last hidden state should already contain all of the useful contextual information in previous time steps. This allows the features extracted from the continuous anchor in the horizontal direction to influence each other. As a result, it is easy to learn relevant features during training to determine whether the anchor is a text.

The internal state of H_t is mapped to the fully connected layer, and the final output layer contains vertical coordinates, scores and boundary-refinement of the proposals. As a result, our integration with the BLSTM layer is elegant, resulting in an efficient end-to-end training model at no additional cost. The model with the BLSTM, the detector can automatically remove candidate areas that are similar to license plates, and weak character proposals can be detected due to fouling or occlusion.

C. Connecting Proposals and Boundary-Refinement

We filter the confidence scores using a threshold of 0.7. Let proposal B_i be a pairing neighbor B_j denoted as $B_j \rightarrow B_i$ if B_i and B_j satisfy: (i) B_j is the horizontal distance closest to B_i, (ii) the distance is less than 50 pixels, (iii) their vertical overlap is greater than 0.7. If $B_j \rightarrow B_i$ and $B_i \rightarrow B_j$, the two proposals are merged. As shown in Fig. 7. If finally many non-license bounding boxes are obtained, we will remove most of the non-license bounding boxes based on the size and angle of the license plate, and train a classifier to remove the last remaining non-license bounding box. As shown in Fig. 8, this classifier is a binary-classification neural network designed by ourselves. It consists of two convolutional layers and three fully connected layers.

Fig. 7. The second stage of proposed method: left shows fine-scale proposals and right shows connecting filtered proposals.

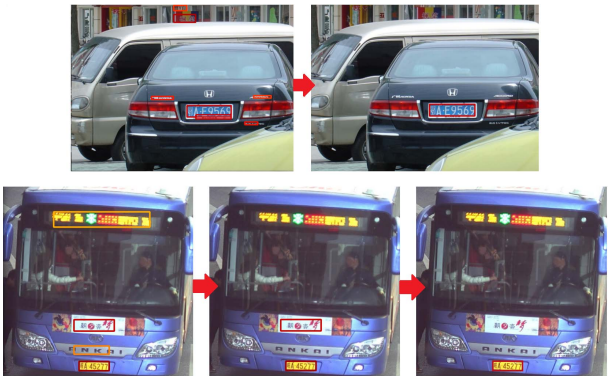

Fig. 8. Top row: remove the non-license bounding boxes by the size and angle; bottom row: remove the non-license bounding boxes by the size, angle and the classifier.

The fine-scale detectors with BLSTM can accurately predict localization in the vertical direction. In the horizontal direction, the image is divided into a series of identical fixed width proposals. This may lead to inaccurate localization when the two horizontal aspects of the character proposal are not completely covered by the terrestrial truth character area, or some boundary proposals are discarded (e.g., with low confidence scores). We need to accurately estimate the offset for each anchor/proposal in both left and right horizontal boundarys (referred as boundary-anchor or boundary-proposal). This inaccurate may not detect the left and right edges of the license plate.

To eliminate this offset, we calculate the relative offset as

$$x = (x_b - m_x^a)/w^a, \quad x^* = (x_b^* - m_x^a)/w^a \tag{11}$$

where x_b is the predicted x-coordinate of the nearest horizontal boundary (e.g., left or right boundary) to current anchor. x_b^* is the ground truth (GT) boundary coordinate in x-axis, which is calculated from the GT bounding box and anchor location. m_x^a is the midpoint of the anchor in x-axis. w_a is the fixed width of the anchor. The boundary-proposals are defined as the start and end proposals when we connect a sequence of fine-scale character proposals into a whole license plate. We use the offsets of the boundary-proposals to refine the final license plate bounding box.

D. Overall Model and Optimization

The proposed method outputs confidence scores (**s**), vertical coordinates (**y**) and boundary-refinement offset (**x**), as shown in Fig. 3. The k predicted anchors are calculated from each

spatial location in the convolution_5 layer and generate $2k$, $2k$ and k parameters through the output layer, respectively.

Following the multi-task loss applied in [23]–[25], we design the overall target loss function (L) for an image as

$$L(\mathbf{s_i}, \mathbf{y_i}, \mathbf{x_i}) = \frac{1}{N}\sum_i L_s^{cl}(\mathbf{s_i}, \mathbf{s_i^*}) + \frac{\lambda_1}{N_y}\sum_j L_y^{reg}(\mathbf{y_j}, \mathbf{y_j^*}) + \frac{\lambda_2}{N_x}\sum_k L_x^{reg}(\mathbf{x_k}, \mathbf{x_k^*}) \quad (12)$$

where L_s^{cl}, L_y^{reg} and L_y^{reg} compute errors of confidence score, y-coordinate and boundary-refinement, respectively. Each anchor is a training sample, and i represents the index of an anchor in a mini-batch. $\mathbf{s_i}$ refers to the predicted probability of anchor i being a true character. We define a valid anchor when the anchor is a defined positive anchor or has Intersection-over-Union (IoU) overlap with a ground truth text proposal greater than 0.5. if the anchor is the positive sample, the $\mathbf{s_i^*}$ is 1, otherwise the $\mathbf{s_i^*}$ is 0. j represents the index of an anchor in the group of valid anchors. $\mathbf{y_j}$ is the prediction y-coordinates associated with the anchor j. k is the index of a side-anchor, which is defined as a set of anchors within a horizontal distance to the left or right boundary of a ground truth license plate bounding box. $\mathbf{x_k}$ is the predicted offsets in x-axis associated to the anchor k. $\mathbf{s_i^*}$, $\mathbf{y_j^*}$ and $\mathbf{x_k^*}$ are ground truth. L_s^{cl} is the classification loss and L_y^{reg} and L_x^{reg} are the bounding boxes regression loss.

Our ultimate goal is to solve

$$G = \underset{\mathbf{s_i}, \mathbf{y_i}, \mathbf{x_i}}{argmin}\, L(\mathbf{s_i}, \mathbf{y_i}, \mathbf{x_i}) \quad (13)$$

The loss function is chosen as mean square error (MSE), and mini-batch gradient descent is used to find the best parameters of the network. The gradients in the convolutional layers are calculated by the back-propagation algorithm, and gradients in the recurrent layers are calculated by the back-propagation through time (BPTT) algorithm [26].

The essence of the method proposed in this paper is to divide the complete license plate into a series of small anchors, and then determine whether these anchors are part of the characters, so for this binary classifier, we need to mark positive and negative labels before training. Positive labels are defined as: (i) an anchor that has an IoU overlap with any GT box greater than 0.7; or (ii) the anchor with the highest IoU overlap with a GT box. The negative anchors are defined as the IoU overlap with all GT boxes less than 0.5. The training labels for the y-coordinate regression ($\mathbf{y}^*$) and offset regression ($\mathbf{x}^*$) are computed as Eq. (2) and (4) respectively.

IV. Experiments

A. Datasets

The images of our data sets in the experiments consist of two categories.

(1) images which have characters without license plates. Because the proposed method locates the license plates by the characters, the character as training samples can be more accurate locating license plates. Our model was trained on 3000 natural images with characters. A part of these pictures are collected by us, others are synthesized [27], and we manually mark the bounding boxes of the characters in the images.

Fig. 9. Sample Chinese license plates images in test set.

TABLE II
The Test Set of License Plates Images

Country	Canada	Malaysia	Taiwan	Thailand	US
Number	135	150	112	75	127
Country	Hong Kong	Europe	Croatia	Chinese	
Number	141	718	200	1680	

(2) images containing license plates. We also collected 3000 images with license plates in the natural scene to train our baseline models. In order to evaluate our proposed method, we selected 1680 Chinese license plates in different scenarios as the test set. Among them, 273 pictures are collected by ourselves, including license plates with various backgrounds and various angles in different scenes. The maximal picture size is 1920*1080, and the minimal size is 250*97. The size of the license plates in the image also varies, and the average is 97*29. The smallest license plate covers three-thousandths of the entire picture. Some examples are shown in Fig 9. In addition, the remaining 1407 images are the license plate data set[1] provided by Sun Yat-sen University. These test images contain license plates for various vehicles and some defaced license plates, the image size is 1600*1200. The data set [28] contains English vehicles plates is publicly available over the Internet. The Caltech license plates dataset[2] contains 127 US vehicles plate images, the image size is 896*592. We also collected multi-country license plates, such as Canada, Malaysia, Thailand, and so on. The statistics of test set are shown in Table II. We also shared our collected test data[3] online.

B. Baselines

We compare our method with several popular methods [5], [20], [29]–[32]: (1) deep learning methods: Faster R-CNN + ZF, Faster R-CNN + VGG_CNN_M_1024,

[1] $https://yun.baidu.com/s/1o79V2ps$

[2] $http://www.vision.caltech.edu/ImageDatasets/cars_markus/cars_markus.tar$

[3] $https://pan.baidu.com/s/1NV7G7_e3PDeGXtc-uGR0rw$

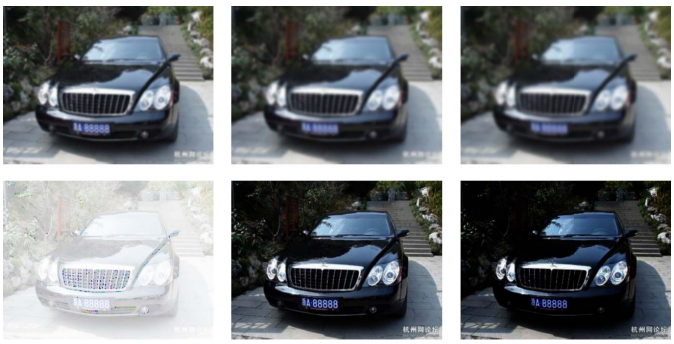

Fig. 10. Images with gaussian blur and gamma transformation. Top row: gaussian blur images with template size of 7, 9, and 11; bottom row: gamma transformation images with gamma parameters of 0.1, 1.6 and 2.5.

Faster R-CNN + VGG16, Ref [20] YOLO-finetunning and SSD-finetunning; (2) traditional methods: CMSER&Sobel, Color&Sobel, CMSER&Color. The evaluation criteria include recall [33] and precision. Recall is as the number of correctly detected license plates divided by the total number of real regions. Precision is as the number of correctly detected license plates divided by the total number of detected regions.

C. Experimental Results and Analysis

1) Robustness: To analyze the robustness of the proposed method given illumination changes and blurred environments, we distort 1680 license plates with gamma transformation and Gaussian blur, as shown in the Fig. 10. The precision of different methods under the condition of illumination change is shown in Fig. 11 (a). SSD-finetunning can achieve good results in general object detection, but the detection effect on small objects is not good, and it can be seen from the table that the robustness is poor. When the gamma is less than 0.4, the precision of Faster R-CNN and YOLO-finetunning volatility is large. When the gamma is in the range of 0.5 to 2.5, the precision of our method still exceeds that of the other methods. The proposed method works robustly regardless of the illumination variation.

Fig. 11 (b) illustrates the change of precision rate with respect to a different degree of blur. The overall curves depict that the value of our method is higher than the other methods. The change of precision is slow as the template scale increases. Fig. 10 (Top) shows when the template scale is 11, the general locating method faces difficulty to detect the license plate; yet the precision of our method achieves 91%.

General license plate localization methods face difficulty in cases of fuzzy and uneven illumination. Our approach overcomes these difficulties, and demonstrates robustness.

2) Localization of Obscured or Defaced License Plates: The framework designed in this paper is not only able to detect the low-quality pictures, but also valid for the obscured and defaced license plates. We found that the obscured and defaced position is on the edge of the license plates and not cover the entire character, our method accurately locates the license plates, as shown in Fig. 12. So our experiment is mainly for the case of the license plate in the vertical direction was completely blocked or defaced which make license plate break.

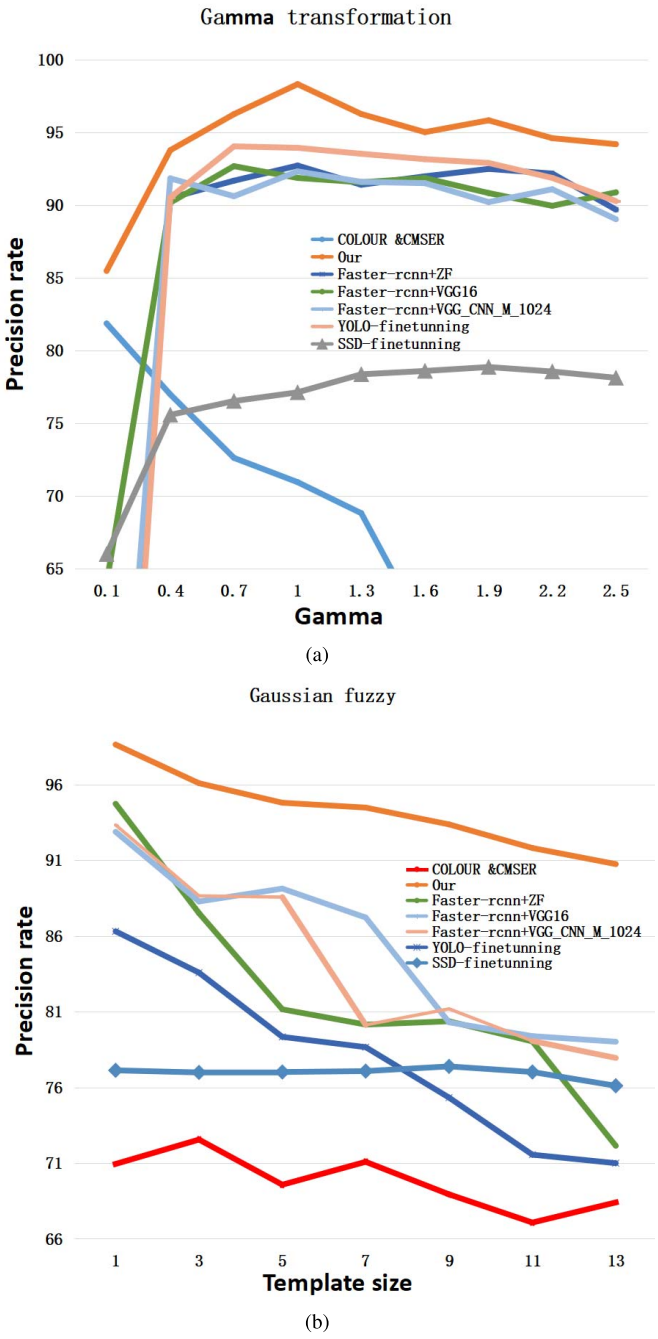

Fig. 11. (a) Precision curves for Gamma transformation images. (b) Precision curves for Gaussian fuzzy images.

Fig. 12. Example images of the license plates with obscured or defaced edge.

The license plate is divided into two parts, which would bring great difficulties to locate.

a) Obscured license plates: Here we manually made 1680 Chinese license plates into obstructed license plates by distorting part of the license plate. The experimental result is

Fig. 13. Top row: example images of the obscured license plates being detected; bottom row: example images of the defaced license plates being detected.

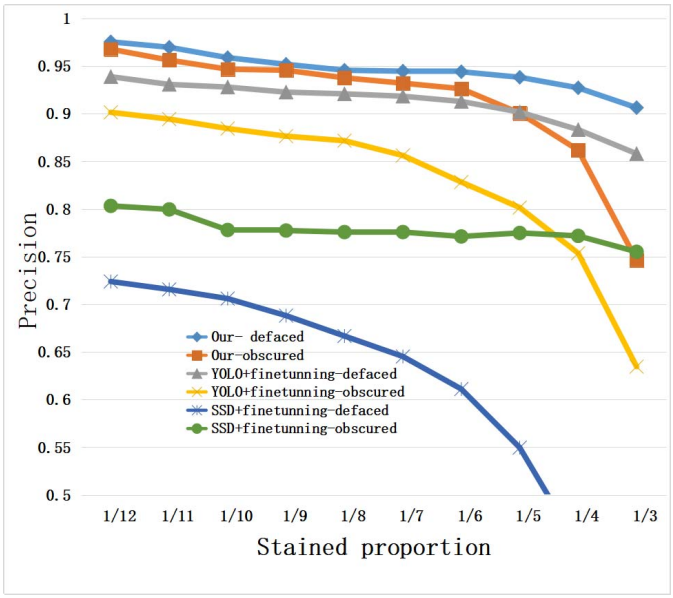

Fig. 14. Precision curves for obscured and defaced license plates.

Fig. 15. The images show the obscured and defaced license plates that have been detected.

Fig. 16. The images show the license plates of different countries that have been detected.

shown in Fig. 13 (Top). We get the precision of the obscured license plates are displayed in Fig. 14. From the orange curve in the figure we can see when the obscured proportions less than 1/7 has high precision, and even if the blocking ratio reaches 1/5, our precision is still up to 91%.

b) Defaced license plates: Because the defaced position of license plates did not lose all the character information. We deal the license plate with high intensity Gaussian blur, which makes the defaced position only retain very little character information. The experimental result is as shown in Fig. 13 (Bottom). We get the precision of the defaced license plates are displayed in Fig. 14. From the blue curve can be seen that the license plate with defaced proportion less than 1/4 still has high precision.

It is demonstrated that our method is more effective than the others in the detection of obscured and defaced license plates. Our detector successfully detects license plates with obscured part less than 1/5 or defaced part less than 1/4. However, the synthetic defaced and obscured license plates are different from the naturally defaced license plates. To further validate the performance of our proposed method, we collected some samples of the naturally obscured and defaced license plates in experiments. It can be seen from Fig. 15 that our proposed method is able to detect such license plates, as well as deformed license plates.

3) Locating License Plates From Different Countries: In order to verify that the trained model can locate different forms of license plate, we collect a lot of license plates from different countries. From Fig.16 can be seen that our method has well localization effect for different countries license plates. The test results of each model are shown in Table III. From the table can be seen when the license plate and the mainland license plate are similar, the precision of other methods is also high, such as Hong Kong. On the contrary, the locating rate is quite poor, such as Canada and so on. The average precision of our proposed method on license plates of different countries is greater than 97%. The highest performance is achieved for Hong Kong license plates as 98.78%. Furthermore, from the last column of Table III,

TABLE III
PRECISION FOR LICENSE PLATES OF DIFFERENT COUNTRIES

Method	Canada	Malaysia	Taiwan	Thailand	Hong Kong	Europe	Croatia	US	Average	Time(ms)
Faster R-CNN+ZF	48.09%	67.69%	63.53%	72.60%	85.00%	68.74%	70.37%	61.38%	66.89%	61
Faster R-CNN+_CNN_M_1024	57.66%	72.89%	79.39%	87.15%	91.30%	74.42%	75.69%	69.80%	75.93%	125
Faster R-CNN+VGG16	51.26%	69.85%	69.86%	83.12%	90.39%	71.36%	74.63%	73.71	72.29%	176
Our method	**94.49%**	**96.45%**	**97.96%**	**96.31%**	**98.71%**	**97.47%**	**97.83%**	**98.41%**	**97.19%**	113

TABLE IV
PRECISION FOR CHINESE LICENSE PLATES

Method	Recall	Precision
CMSER&Sobel	79.43%	42.31%
Color&Sobel	77.01%	41.60%
CMSER&Color	82.05%	69.34%
Faster R-CNN+ZF	92.72%	89.17%
Faster R-CNN+VGG_CNN_M_1024	91.87%	86.98%
Faster R-CNN+VGG16	92.32%	89.01%
Ref [20]	81.23%	79.89%
YOLO-finetunning	88.30%	85.04%
SSD-finetunning	83.73%	81.36%
Our method without fine-scale proposals	79.31%	71.58%
Our method	**99.10%**	**98.68%**

Fig. 17. The images show the Chinese license plates have been detected.

we also find that the proposed method does not increase the computation much.

From the experiment, we can know the two-stage method with great practicality, which breaks the limitations of the general detectors can only locate a class of license plates. Some detectors want to locate license plates from different countries which need to train different models with different training sets. However, we only need to train a model which can locate the license plates of different countries.

Finally, we evaluate the proposed method in the test set with 1680 Chinese license plates in different scenarios. The results are shown in Table IV and Fig.17. It can be seen from the table that the approaches with neural network in the license plate localization occupy a great advantage. SSD-finetunning and YOLO-finetunning can achieve good results in general object detection, but not well adapted for the license plate detection. Since our test set contains some fuzzy and uneven illumination license plates, we find from Fig. 11 that SSD-finetunning performs poor for such cases. We see the same result from Table IV. To evaluate the impact of fine-scale proposals on the performance, we compared with the model including only CNN and LSTM without fine-scale proposals in experiments. In this case, the input size of the network needs to be fixed, and the model cannot detect multiple license plates in one image, which affects the detection performance. From the results of Table IV, it can be found that the detector performance is significantly degraded. This shows the importance of fine-scale proposals for the proposed method.

V. CONCLUSION

We have proposed an effective two-stage unconstrained license plate localization method jointing CNN and RNN. We verified on multiple data sets, and extensive experiments demonstrate that the proposed method with CNN-RNN is not only better than the exiting state-of-the-art methods, but also has a good ability to detect obscured or defaced license plate with less false detection. The proposed method achieves a average precision of 97.11% on the data set of foreign license plates and 98.68% on the data set of Chinese license plates. The proposed method is highly applicable to detect challenging license plates such as smeared or obscured ones in real life, or licenses from multiple countries in unconstrained scenarios. However, since the proposed method sets a vertical anchor mechanism according to the characteristics of the license plate, the performance is degraded if the tilt angle is too large. Therefore, in the future work, we will optimize the vertical anchor mechanism to accurately detect license plates with large tilt angles.

REFERENCES

[1] A. C. Roy, M. K. Hossen, and D. Nag, "License plate detection and character recognition system for commercial vehicles based on morphological approach and template matching," in *Proc. Int. Conf. Elect. Eng. Inf. Commun. Technol.*, Sep. 2016, pp. 1–6.

[2] B.-G. Han, J. T. Lee, K.-T. Lim, and Y. Chung, "Real-time license plate detection in high-resolution videos using fastest available cascade classifier and core patterns," *ETRI J.*, vol. 37, no. 2, pp. 251–261, 2015.

[3] A. H. Ashtari, M. J. Nordin, and M. Fathy, "An iranian license plate recognition system based on color features," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1690–1705, Aug. 2014.
[4] G. Kosala, A. Harjoko, and S. Hartati, "License plate detection based on convolutional neural network: Support vector machine (CNN-SVM)," in *Proc. Int. Conf. Video Image Process.*, 2017, pp. 1–5.
[5] 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. 2017.
[6] X. Ma and E. Hovy. (2016). "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF." [Online]. Available: https://arxiv.org/abs/1603.01354
[7] H. Yu and Y. Ko, "Expansion of word representation for named entity recognition based on bidirectional LSTM," *J. KIISE*, vol. 44, no. 3, pp. 306–313, 2017.
[8] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3286–3293.
[9] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, 2005.
[10] 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.
[11] J. Tian, G. Wang, J. Liu, and Y. Xia, "License plate detection in an open environment by density-based boundary clustering," *J. Electron. Imag.*, vol. 26, no. 3, pp. 033017-1–033017-11, 2017.
[12] M. Jie, H. Li, and J.-J. Liu, "Detection algorithm of cascaded Adaboost license plate based on HSV color model and MB_LBP features," *J. Sichuan Univ. (Natural Sci. Ed.)*, vol. 55, no. 2, p. 13, Mar. 2018.
[13] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
[14] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[16] F.-C. Chen and R. M. R. Jahanshahi, "NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4392–4400, May 2018.
[17] H. Li and C. Shen. (2016). "Reading car license plates using deep convolutional neural networks and LSTMs." [Online]. Available: https://arxiv.org/abs/1601.05610
[18] 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*, Oct. 2017, pp. 55–62.
[19] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, "Segmentation- and annotation-free license plate recognition with deep localization and failure identification," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2351–2363, Sep. 2017.
[20] T. Ying, L. Xin, and L. Wanxiang, "License plate detection and localization in complex scenes based on deep learning," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 6569–6574.
[21] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," in *Proc. 30th AAAI Conf. Artif. Intell. (AAAI)*, vol. 116, no. 1, 2015, pp. 3501–3508.
[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[23] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.
[24] X. Zhang, F. Zhou, Y. Lin, and S. Zhang, "Embedding label structures for fine-grained feature representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1114–1123.
[25] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2529–2541, Jun. 2016.
[26] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.
[27] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. (2014). "Synthetic data and artificial neural networks for natural scene text recognition." [Online]. Available: https://arxiv.org/abs/1406.2227
[28] 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.
[29] S. Israni and S. Jain, "Edge detection of license plate using sobel operator," in *Proc. Int. Conf. Elect., Electron., Optim. Techn. (ICEEOT)*, Mar. 2016, pp. 3561–3563.
[30] W. Wang, Q. Jiang, X. Zhou, and W. Wan, "Car license plate detection based on MSER," in *Proc. Int. Conf. Consum. Electron., Commun. Netw. (CECNet)*, Apr. 2011, pp. 3973–3976.
[31] X. Yang, Y. Zhao, J. Fang, Y. Lu, Y. Zhang, and Y. Yuan, "A license plate segmentation algorithm based on MSER and template matching," in *Proc. 12th Int. Conf. Signal Process. (ICSP)*, Oct. 2014, pp. 1195–1199.
[32] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[33] X. Sun, P. Wu, and S. C. H. Hoi, "Face detection using deep learning: An improved faster RCNN approach," *Neurocomputing*, vol. 299, pp. 42–50, Jul. 2018.

Jingjing Zhang, photograph and biography not available at the time of publication.

Yuanyuan Li, photograph and biography not available at the time of publication.

Teng Li, photograph and biography not available at the time of publication.

Lina Xun, photograph and biography not available at the time of publication.

Caifeng Shan, photograph and biography not available at the time of publication.