Toward End-to-End Car License Plate Detection and Recognition With Deep Neural Networks

Hui Li, Peng Wang, and Chunhua Shen

***Abstract*— In this paper, we tackle the problem of car license plate detection and recognition in natural scene images. We propose a unified deep neural network, which can localize license plates and recognize the letters simultaneously in a single forward pass. The whole network can be trained end-to-end. In contrast to existing approaches which take license plate detection and recognition as two separate tasks and settle them step by step, our method jointly solves these two tasks by a single network. It not only avoids intermediate error accumulation but also accelerates the processing speed. For performance evaluation, four data sets including images captured from various scenes under different conditions are tested. Extensive experiments show the effectiveness and the efficiency of our proposed approach.**

***Index Terms*— Car plate detection and recognition, convolutional neural networks, recurrent neural networks.**

I. Introduction

AUTOMATIC car license plate detection and recognition plays an important role in intelligent transportation systems. It has a variety of potential applications ranging from security to traffic control, and attracts considerable research attentions during recent years.

However, most of the existing algorithms only work well either under controlled conditions or with sophisticated image capture systems. It is still a challenging task to read license plates accurately in an uncontrolled environment. The difficulty lies in the highly complicated backgrounds, like the general text in shop boards, windows, guardrail or bricks, and random photographing conditions, such as illumination, distortion, occlusion or blurring.

Previous work on license plate detection and recognition usually considers plate detection and recognition as two separate tasks, and solves them respectively by different methods. However, the tasks of plate detection and recognition are highly correlated. Accurate bounding boxes obtained via detection methods can improve the recognition accuracy, while the recognition result can be used to eliminate false positives vice versa. Thus in this paper, we propose a unified framework to jointly tackle these two tasks at the same level. A deep neural network is designed, which takes an image as input and outputs the locations of license plates as well as plate labels simultaneously, with both high efficiency and accuracy. We prove that the low level features can be used for both detection and recognition. The whole network can be trained end-to-end, without using any heuristic rule. An overview of the network architecture is shown in Figure 1. To our knowledge, this is the first work that integrates both license plate detection and recognition into a single end-to-end trainable network and solves them at the same time. The main contributions of this work are as follows:

- A single unified deep neural network is proposed, which can detect license plates from an image and recognize the labels all at once. The whole framework involves no heuristic processes, such as the use of plate colors or character space, and avoids intermediate procedures like character grouping or separation. It can be trained end-to-end, with only the image, plate positions and labels needed for training. The resulting system achieves high accuracy on both plate detection and letter recognition.
- Secondly, the convolutional features are shared by both detection and recognition, which leads to fewer parameters compared to using separated models. Moreover, with the joint optimization of both detection and recognition losses, the extracted features would have richer information. Experiments show that both detection and recognition performance can be boosted via using the jointly trained model.
- By integrating plate recognition directly into the detection pipeline, instead of addressing them by separate models, the resulting system is more efficient. With our framework, we do not need to crop the detected license plates from the input image and then recognize them by a separate network. The whole framework takes about 0.31 second for an input image of 600×600 pixels on a Titan X GPU.

It should be note that although a number of methods has been proposed for both text detection and recognition in natural scene, e.g., [1]–[8], our method is quite different from them. The most remarkable difference is that our network can be trained end-to-end, while other methods combine results from separately trained models to obtain the final detection and recognition results. With this innovation, some pre-processing, like character detection or character grouping, are eliminated, and the intermediate errors can be avoided. The learned features can be more discriminative and lead to a better

Manuscript received April 5, 2017; revised September 26, 2017 and March 24, 2018; accepted May 15, 2018. Date of publication August 2, 2018; date of current version February 28, 2019. The Associate Editor for this paper was D. Fernandez-Llorca. *(Corresponding author: Peng Wang.)*

H. Li and C. Shen are with the School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia.

P. Wang is with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: peng.wang@nwpu.edu.cn).

Digital Object Identifier 10.1109/TITS.2018.2847291

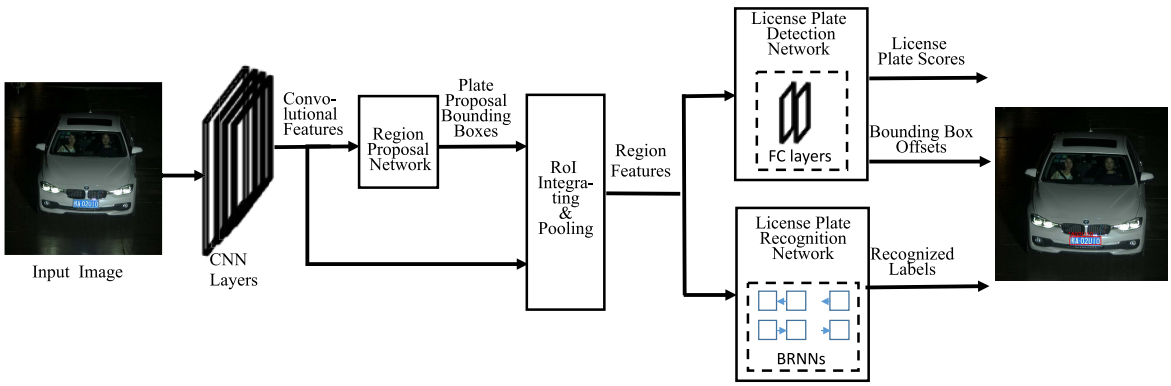

Fig. 1. The overall structure of our model. It consists of several convolutional layers, a region proposal network for license plate proposals generation, proposal integrating and pooling layer, multi-layer perceptrons for plate detection and bounding box regression, and RNNs for plate recognition. Given an input RGB image, with a single forward evaluation, the network outputs scores of predicted bounding boxes being license plates, bounding box offsets with a scale-invariant translation and log-space height/width shift relative to a proposal, as well as the recognized license plate labels at the same time. The extracted region features are used by both detection and recognition, which not only shares computation, but also reduces model size.

performance. We will investigate some related work in the following section.

The rest of the paper is organized as follows. Section 2 gives a brief discussion on related work. Section 3 presents the integrated model, and introduces each part in detail. Experimental verifications are followed in Section 4, and conclusions are drawn in Section 5.

II. Related Work

As license plate detection and recognition are generally addressed separately, we give a brief introduction to previous work on each aspect respectively. In addition, we also review some related work on end-to-end scene text detection and recognition, which is quite close to our topic.

1) License Plate Detection: License plate detection aims to localize the license plates in the image in the form of bounding boxes. Existing methods can be roughly classified into four categories [9]–[11]: edge-based, color-based, texture-based, and character-based.

License plates are normally in a rectangular shape with a specific aspect ratio, and they present higher edge density than elsewhere in the image, so edge information is used widely to detect license plates. In [12] an edge-based method was developed for plate detection. Expectation Maximization (EM) was applied for edge clustering which extracts the regions with dense sets of edges and with shapes similar to plates as the candidate license plates. In [13], a novel line density filter approach was proposed to connect regions with high edge density and remove sparse regions in each row and column from a binary edge image. Edge-based methods are fast in computation, but they cannot be applied to complex images as they are too sensitive to unwanted edges.

Color-based approaches are based on the observation that color of the license plate is usually different from that of the car body. In [14], a plate detection method was developed by analyzing the target color pixels. A color-geometric template was utilized to localize Iranian license plates via strip search. Chang *et al.* [15] proposed a method to detect Taiwan license plates in RGB images based on the different foreground and background colors. They developed a color edge detector which is sensitive to black-white, red-white and green-white edges. Color-based methods can be used to detect inclined or deformed license plates. However, they are very sensitive to various illumination conditions in natural scene images, and they cannot distinguish other objects in the image with similar color and size as the license plates.

Texture-based approaches attempted to detect license plates according to the unconventional pixel intensity distribution in plate regions. Yu *et al.* [16] used a wavelet transform at first to get the horizontal and vertical details of an image. Empirical Mode Decomposition (EMD) analysis was then employed to deal with the projection data and locate the desired wave crest which indicates the position of a license plate. Giannoukos *et al.* [17] developed a Sliding Concentric Window (SCW) algorithm to identify license plates based on the local irregularity property of plate texture in an image. Operator Context Scanning (OCS) was proposed to accelerate the detection speed. Texture-based methods use more discriminative characteristics than edge or color, but result in a higher computational complexity.

Considering the fact that license plates consist of a string of characters, much work appeared recently based on the character-based feature as it includes more specific information. Zhou *et al.* [10] formulated license plate detection as a visual matching problem. Principal Visual Word (PVW) was generated for each character which contains geometric clues such as orientation, characteristic scale and relative position, and used for plate extraction. Li *et al.* [18] applied Maximally Stable Extremal Region (MSER) at the first stage to extract candidate characters in images. Conditional Random Field (CRF) was then constructed to represent the relationship among license plate characters. License plates were finally localized through the belief propagation inference on CRF. Llorca *et al.* [19] made use of a combination of the MSER and Stroke Width Transform (SWT) to detect isolate character regions. The license plates were finally bordered using the probabilistic Hough transform. Character-based methods are more reliable and can lead to a high recall. However, the

performance is affected largely by other text in the image background.

2) License Plate Recognition: Previous work on license plate recognition typically segments characters in the license plate firstly, and then recognizes each segmented character using Optical Character Recognition (OCR) techniques. For example, in [20], Extremal Regions (ER) were employed to segment characters from coarsely detected license plates and to refine plate location. Restricted Boltzmann machines were applied to recognize the characters. In [12], MSER was adopted for character segmentation. Local Binary Pattern (LBP) features were extracted and classified using a Linear Discriminant Analysis (LDA) classifier for character recognition. Hou *et al.* [21] proposed to segment license plate character based on SWT, which can process rotated license plates.

However, character segmentation by itself is a really challenging task that is prone to be influenced by uneven lighting, shadow and noise in the image. It has an immediate impact on plate recognition. The plate cannot be recognized correctly if the segmentation is improper, even if we have a strong recognizer. With the development of deep neural networks, approaches were proposed to recognize the whole license plate directly without character separation. Goodfellow *et al.* [22] proposed to train a probabilistic model to read arbitrary multi-digit numbers without individual character localization or segmentation, using a large scale distributed deep neural network. In [23], segmentation and optical character recognition were jointly performed using Hidden Markov Models (HMMs) where the most likely label sequence was determined by Viterbi algorithm. In [24], plate recognition was regarded as a sequence labeling problem. Convolutional Neural Networks (CNNs) was employed in a sliding window manner to extract a sequence of feature vectors from the license plate bounding box. Recurrent Neural Networks (RNNs) with Connectionist Temporal Classification (CTC) [25] were adopted to label the sequential data without character separation.

3) End-to-End Scene Text Detection and Recognition System: As a similar task, plenty of approaches have been proposed for general text detection and recognition in natural scene images. For instance, Neumann and Matas [1], [2] proposed methods for both text localization and recognition in real-word images. A character detection step was conducted firstly by MSER in [1] and ER in [2] respectively. Character grouping was followed in a heuristic manner. Word localization and recognition were achieved by combining the characterness scores with a given dictionary. Wang *et al.* [3] proposed a word spotting method based on a given lexicon. A multi-scale sliding window based method was employed firstly to localize character candidates, using random ferns classifiers. Pictorial Structures (PS) formulation was adopted then that takes the locations and scores of detected characters as input to find an optimal configuration of a particular word from an given lexicon, and outputs both word bounding boxes and word labels simultaneously. Neumann and Matas [26] proposed an efficient algorithm to select the optimal sequence of characters from a directed graph, which is constructed with character classification scores, character intervals and language priors. The sequence of regions and their labels induced by the optimal path are the outputs. In these methods, a separate character detection step is always needed. PhotoOCR [5] combined the outputs of three different text detection approaches to get a high text line detection recall. Recognition began with over-segmentation of text line to identify candidate character regions, and was achieved by combining character classifiers and language model likelihoods. Jaderberg *et al.* [7] used a combination of complementary proposal generation techniques for word region proposal, and a random forest classifier for word/no-word classification. Word recognition was achieved by training a deep CNN over a huge dictionary of 90k words. Liao *et al.* [8] combined "TextBoxes" and "CRNN" for word detection and recognition separately, where "TextBoxes" is a 28-layer fully convolutional network which is trained to produce word level bounding boxes directly from input images, while "CRNN" [27] is a combination of CNN, RNN and CTC loss in one neural network for the image-based sequence recognition task. In these methods text detection and recognition are achieved by two separately trained models. The most similar framework to ours' is [28], where an end-to-end trainable scene text localization and recognition framework was proposed. However, in this method, the detected text needed to be cropped out from the image and recognized by another CNN model. It is only the detection and recognition losses that jointly optimized. The features need to be calculated again in the recognition network. By contrast, in our framework, the low-level convolutional features are shared for both detection and recognition modules, which can reduce the parameters and save computational time.

III. Model

Different from the above-mentioned methods, our approach addresses both detection and recognition using a single deep neural network. As illustrated in Figure 1, our model consists of a number of convolutional layers to extract discriminative features for license plates, a region proposal network tailored specifically for car license plates, a Region of Interest (RoI) pooling layer, multi-layer perceptrons for plate detection and bounding box regression, and RNNs with CTC for plate recognition. With this architecture, the plate detection and recognition can be achieved simultaneously, with one network and a single forward evaluation of the input image. Moreover, the whole network is trained end-to-end, with both localization loss and recognition loss being jointly optimized, and shows improved performance. In the following subsections, we give a detailed description about each component.

A. Model Architecture

1) Low-Level Feature Extraction: The VGG-16 network [29] which is pre-trained on ImageNet [30] is adopted here to extract low level CNN features. VGG-16 consists of 13 layers of 3×3 convolutions followed by Rectified Linear Unit (ReLU) non-linearity, 5 layers of 2×2 max-pooling, and fully connected layers. Here we keep all the convolutional layers and abandon the fully connected layers as we require local features at each position for plate detection. Given that

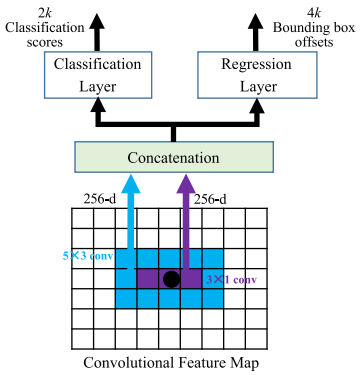

Fig. 2. Plate Proposal Generation. Two rectangular convolutional filters are applied in each sliding window, which include rich contextual information. The features are concatenated and then fed into the classification layer for plate/non-plate classification and the regression layer to calculate coordinate offsets, with respect to k anchors at each position.

the license plates are small compared with the whole image size, we use 2 pooling layers instead of 5, in case the feature information of license plates is vanished after pooling. So the resulting feature maps are one fourth size of the original input image. The higher-resolution feature maps will benefit the detection of small objects [31]. They are used as a base for both detection and recognition.

2) Plate Proposal Generation: Ren *et al.* [32] designed a Region Proposal Network (RPN) for object detection, which can generate candidate objects in images. RPN is a fully convolutional network which takes the low-level convolutional features as input, and outputs a set of potential bounding boxes. It can be trained end-to-end so that high quality proposals can be generated. In this work, we modify RPN slightly to make it suitable for car license plate proposal.

According to the scales and aspect ratios of license plates in our datasets, we designed 6 scales (the heights are respectively 5, 8, 11, 14, 17, 20) with an aspect ratio (width/height = 5), which results in $k = 6$ anchors at each position of the input feature maps. In addition, inspired by inception-RPN [33], we use two 256-d rectangular convolutional filters ($W_1 = 5$, $H_1 = 3$ and $W_2 = 3, H_2 = 1$) instead of the regularly used square one, as shown in Figure 2. The two convolutional filters are applied simultaneously across each sliding position. The extracted local features are concatenated along the channel axis and form a 512-d feature vector, which is then fed into two separate fully convolutional layers for plate/non-plate classification and box regression. On the one hand, these rectangle filters are more suitable for objects with larger aspect ratios (license plates). On the other hand, the concatenated features keep both local and contextual information, which will benefit the plate classification.

For k anchors at each sliding position on the feature map, the plate classification layer outputs $2k$ scores which indicate the probabilities of the anchors as license plates or not. The bounding box regression layer outputs $4k$ values which are the offsets of anchor boxes to a nearby ground-truth. Given an anchor with the center at (x_a, y_a), width w_a and height h_a, the regression layer outputs 4 scalars (t_x, t_y, t_w, t_h) which are the scale-invariant translation and log-space height/width shift. The bounding box after regression is given by

$$x = x_a + t_x w_a, y = y_a + t_y h_a,$$
$$w = w_a \exp(t_w), h = h_a \exp(t_h),$$

where x, y are the center coordinates of the bounding box after regression, and w, h are its width and height.

For a convolutional feature map with size $M \times N$, there will be $M \times N \times k$ anchors in total. Those anchors are redundant and highly overlapped with each other. Moreover, there are much more negative anchors than positive ones, which will lead to bias during training if we use all those anchors. We randomly sample 256 anchors from one image as a mini-batch, where the ratio between positive and negative anchors is up to 1:1. The anchors that have Intersection over Union (IoU) scores larger than 0.7 with any ground-truth bounding box are selected as positives, while anchors with IoU lower than 0.3 as negatives. The anchors with the highest IoU scores are also regarded as positives, so as to make sure that every ground-truth box has at least one positive anchor. If there are not enough positive anchors, we pad with negative ones.

The binary logistic loss is used here for box classification, and smooth L_1 loss [32] is employed for box regression. The multi-task loss function used for training RPN is

$$L_{RPN} = \frac{1}{N_{cls}} \sum_{i=1}^{N_{cls}} L_{cls}(p_i, p_i^\star) + \frac{1}{N_{reg}} \sum_{i=1}^{N_{reg}} L_{reg}(\mathbf{t}_i, \mathbf{t}_i^\star), \quad (1)$$

where N_{cls} is the size of a mini-batch and N_{reg} is the number of positive anchors in this batch. Bounding box regression is only for positive anchors, as there is no ground-truth bounding box matched with negative ones. p_i is the predicted probability of anchor i being a license plate and $p_i^\star$ is the corresponding ground-truth label (1 for positive anchor, 0 for negative anchor). $\mathbf{t}_i$ is the predicted coordinate offsets $(\mathbf{t}_{i,x}, \mathbf{t}_{i,y}, \mathbf{t}_{i,w}, \mathbf{t}_{i,h})$ for anchor i, and $\mathbf{t}_i^\star$ is the associated offsets for anchor i relative to the ground-truth. RPN is trained end-to-end with back-propagation and Stochastic Gradient Descent (SGD).

At test time, the forward evaluation of RPN will result in $M \times N \times k$ anchors with objectiveness scores as well as bounding box offsets. We employ Non-Maximum Suppression (NMS) to select 100 proposals with higher confidences based on the predicted scores for the following processing.

3) Proposal Processing and Pooling: As we state before, 256 anchors are sampled from the $M \times N \times k$ anchors to train RPN. After bounding box regression, the 256 samples will later be used for plate detection and recognition.

We denote the bounding box samples as $p = (x^{(1)}, y^{(1)}, x^{(2)}, y^{(2)})$, where $(x^{(1)}, y^{(1)})$ is the top-left coordinate of the bounding box, and $(x^{(2)}, y^{(2)})$ is the bottom-right coordinate of the bounding box. For all the positive proposals $p_{i,j} = (x_{i,j}^{(1)}, y_{i,j}^{(1)}, x_{i,j}^{(2)}, y_{i,j}^{(2)})$, $i = 1, \ldots, n$ that are associated with the same ground-truth plate g_j, a bigger bounding box $b_j = (x_j^{(1)}, y_j^{(1)}, x_j^{(2)}, y_j^{(2)})$ is constructed that encompasses all

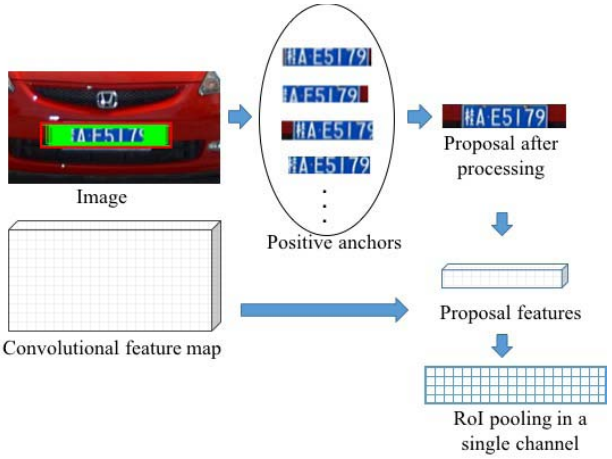

Fig. 3. Proposal processing and pooling. Since some positive anchors (the green ones in the image) can not cover all letters in the license plate, the corresponding features may not be included after RoI pooling, which is not suitable for recognition. Hence we construct a new proposal (the red one in the image) that encircles all positive anchors and is big enough to contain sufficient features for recognition. RoI pooling is then followed to get fixed size feature maps. It should be note that this process is only implemented in training stage, while in test stage, the select 100 proposals with higher plate classification scores are processed directly for detection and recognition.

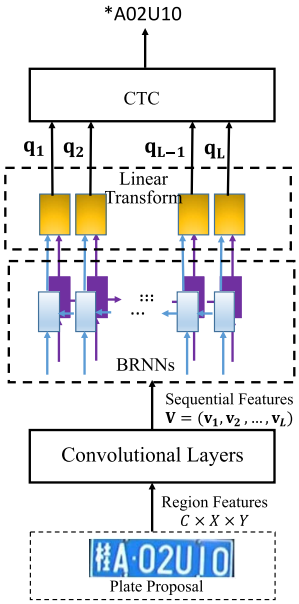

Fig. 4. Plate Recognition Network. The pooled region features are regarded as a feature sequence, and encoded by BRNNs, which capture the context information in both sides. CTC are used for plate decoding without character separation.

proposals $p_{i,j}$, *i.e.*,

$$x_j^{(1)} = \min_{i=1,\ldots,n}(x_{i,j}^{(1)}),\ y_j^{(1)} = \min_{i=1,\ldots,n}(y_{i,j}^{(1)}),$$
$$x_j^{(2)} = \max_{i=1,\ldots,n}(x_{i,j}^{(2)}),\ y_j^{(2)} = \max_{i=1,\ldots,n}(y_{i,j}^{(2)}).$$

This process is also illustrated in Figure 3. The constructed bounding boxes b_j, $j = 1, \ldots, m$ will then be used as positive samples for later plate detection and recognition. To avoid the bias caused by the unbalanced distribution between positive and negative samples, we randomly choose $3m$ negative ones from the 256 samples and form a mini-batch with $4m$ samples.

Considering that the sizes of the samples are different from each other, in order to interface with the plate detection network as well as the recognition network, RoI pooling [34] is adopted here to extract a fixed-size feature representation. Each RoI is projected into the image convolutional feature maps, which results in feature maps of size $H' \times W'$. The varying sized feature maps $H' \times W'$ are then divided into $X \times Y$ grids, where boundary pixels are aligned by rounding. Features are max-pooled within each grid. Here we choose $X = 4$ and $Y = 28$ instead of 7×7 that is used in [34], because of the subsequent plate recognition task. To be specific, since we need to recognize each character in the license plate, it would be better if we keep more feature horizontally. However, the model size p from this layer to the next fully connected layer is closely related to X and Y, *i.e.*, $p \propto XY$. A larger feature map size will result in more parameters and increase the computation burden. After experimental analysis, we adopt a longer width $Y = 28$ and a shorter height $X = 4$.

4) Plate Detection Network: Plate detection network aims to judge whether the proposed RoIs are car license plate or not, and refine the coordinates of plate bounding boxes.

Two fully connected layers with 2048 neurons and a dropout rate of 0.5 are employed here to extract discriminative features for license plate detection. The features from each RoI are flattened into a vector and passed through the two fully connected layers. The encoded features are then fed concurrently into two separate linear transformation layers respectively for plate classification and bounding box regression. The plate classification layer has 2 outputs, which indicate the softmax probability of each RoI as plate/non-plate. The plate regression layer produces the bounding box coordinate offsets for each proposal, as used in the region proposal network.

5) Plate Recognition Network: Plate recognition network aims to recognize each character in RoIs based on the extracted region features. To avoid the challenging task of character segmentation, we regard the plate recognition as a sequence labeling problem. Bidirectional RNNs (BRNNs) with CTC loss [25] are employed to label the sequential features, which is illustrated in Figure 4.

The region features after RoI pooling are denoted as $\mathbf{Q} \in \mathbb{R}^{C \times X \times Y}$, where C is the channel size. First of all, we add two additional convolutional layers with ReLUs. Both of them use 512 filters. The kernel sizes are 3 and 2 respectively, with a padding of 1 used in the first convolutional layer. A rectangular pooling window with $k_W = 1$ and $k_H = 2$ is adopted between them, which would be beneficial for recognizing characters with narrow shapes, such as '1' and 'I', referring to [27]. These operations will reform the region features $\mathbf{Q}$ to a sequence with the size as $D \times L$, where $D = 512$ and $L = 19$. We denote the resulting features as $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_L)$, where $\mathbf{v}_i \in \mathbb{R}^D$.

Then BRNNs are applied on top of the sequential features. As presented in Figure 4, two separated RNN layers with 512 units are used. One processes the feature sequence forward, with the hidden state updated via $\mathbf{h}_t^{(f)} = \mathrm{g}(\mathbf{v}_t, \mathbf{h}_{t-1}^{(f)})$.

The other one processes it backward, with the hidden state updated via $\mathbf{h}_t^{(b)} = \mathrm{g}(\mathbf{v}_t, \mathbf{h}_{t+1}^{(b)})$. The two hidden states are concatenated together and fed to a linear transformation with 37 outputs. Softmax layer is followed to transform the 37 outputs into probabilities, which correspond to the distributions over 26 capital letters, 10 digits, and a special non-character class. We record the probabilities at each time step. Hence, after BRNNs encoding, the feature sequence $\mathbf{V}$ is transformed into a sequence of probability estimation $\mathbf{q} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_L)$ with the same length as $\mathbf{V}$. BRNNs capture abundant contextual information from both directions, which will make the character recognition more accurate. To overcome the shortcoming of gradient vanishing or exploding during traditional RNN training, Long-Short Term Memory (LSTM) [35] is employed here. It defines a new cell structure called memory cell, and three multiplicative gates (*i.e.,* input gate, forget gate and output gate), which can selectively store information for a long time.

Then CTC layer [25] is adopted here for sequence decoding, which is to find an approximately optimal path π^* with maximum probability through the BRNNs' output sequence $\mathbf{q}$, *i.e.,*

$$\pi^* \approx \mathcal{B}\left(\arg\max_{\pi} P(\pi|\mathbf{q})\right). \tag{2}$$

Here a path π is a label sequence based on the output activation of BRNNs, and $P(\pi|\mathbf{q}) = \prod_{t=1}^{L} P(\pi_t|\mathbf{q})$. The operator $\mathcal{B}$ is defined as the operation of removing the repeated labels and the non-character label from the path. For example, $\mathcal{B}(a-a-b-) = \mathcal{B}(-aa--a-bb) = (aab)$. Details of CTC can refer to [25]. The optimal label sequence π^* is exactly the recognized plate label.

B. Loss Functions and Training

As we demonstrate previously, the whole network takes as inputs an image, the plate bounding boxes and the associated labels during training time. After we obtain the samples as well as the region features, we combine the loss terms for plate detection and recognition, and jointly train the detection and recognition networks. Hence, the multi-task loss function is defined as

$$L_{DRN} = \frac{1}{N}\sum_{i=1}^{N} L_{cls}(p_i, p_i^\star) + \frac{1}{N_+}\sum_{i=1}^{N_+} L_{reg}(\mathbf{t}_i, \mathbf{t}_i^\star) + \frac{1}{N_+}\sum_{i=1}^{N_+} L_{rec}(\mathbf{q}^{(i)}, \mathbf{s}^{(i)}) \tag{3}$$

where N is the size of a mini-batch used in detection network and N_+ is the number of positive samples in this batch. The definitions of L_{cls} and L_{reg} are the same as that used in RPN. p_i, $p_i^\star$, $\mathbf{t}_i$, $\mathbf{t}_i^\star$ also use the same definition as that used in RPN. $\mathbf{s}^{(i)}$ is the ground-truth plate label for sample i and $\mathbf{q}^{(i)}$ is the corresponding output sequence by BRNNs.

It is observed that the length of BRNNs' outputs $\mathbf{q}^{(i)}$ is not consistent with the length of target label $\mathbf{s}^{(i)}$. Based on CTC loss in [25], the objective function for plate recognition is defined as the negative log probability of the network outputting correct label, *i.e.,*

$$L_{rec}(\mathbf{q}^{(i)}, \mathbf{s}^{(i)}) = -\log P(\mathbf{s}^{(i)}|\mathbf{q}^{(i)}) \tag{4}$$

where

$$P(\mathbf{s}^{(i)}|\mathbf{q}^{(i)}) = \sum_{\pi:\mathcal{B}(\pi)=\mathbf{s}^{(i)}} P(\pi|\mathbf{q}^{(i)}) \tag{5}$$

which is the sum of probabilities of all π that can be mapped to $\mathbf{s}^{(i)}$ by $\mathcal{B}$.

We use the approximate joint training process [32] to train the whole network, ignoring the derivatives with respect to the proposed boxes' coordinates. Fortunately, this does not have a great influence on the performance [32]. We train the whole network using SGD. CNNs for extracting low-level features are initialized from the pre-trained VGG-16 model. We do not fine-tune the first four convolutional layers for efficiency. The rest of CNN layers are fine-tuned only in the first $50K$ iterations. The other weights are initialized according to Gaussian distribution. For optimization, we use ADAM [36], with an initial learning rate of 10^{-5} for parameters in the pre-trained VGG-16 model, and 10^{-4} for other parameters. The latter learning rate is halved every $10K$ iterations until 10^{-5}. The network is trained for $200K$ iterations. Each iteration uses a single image sampled randomly from training dataset. For each training image, we resize it to the shorter side of 700 pixels, while the longer side no more than 1500 pixels.

IV. Experiments

In this section, we conduct experiments to verify the effectiveness of the proposed method. Our network is implemented using Torch 7. The experiments are performed on NVIDIA Titan X GPU with 12GB memory.

A. Datasets

Four datasets are used here to evaluate the effectiveness of our proposed method.

The first dataset is composed of car license plates from China, denoted as CarFlag-Large. We collected 460000 images in total. The images are captured from frontal viewpoint by fixed surveillance cameras under different weather and illumination conditions, *e.g.*, in sunny days, in rainy days, or at night, with a resolution of 1600×2048. The plates are nearly horizontal. Only the nearest license plate in the image is labeled in the ground-truth file. We use 322000 images for training, and 138000 images for test.

The second dataset is the Application-Oriented License Plate (AOLP) database [12]. It has 2049 images in total with Taiwan license plates. This database is categorized into three subsets with different level of difficulty and photographing condition, as refer to [12]: Access Control (AC), Traffic Law Enforcement (LE), and Road Patrol (RP). Since we do not have any other images with Taiwan license plates, to train the network, we use images from different sub-datasets for training and test separately. For example, we use images from LE and RP subsets to train the network, and evaluate the performance on AC subset. Considering the small number of

TABLE I
DATASET SUMMARY

Dataset		Image quantity	Image resolution (height × width)	Number of training/test Images	Number of plate	Plate size (height × width)
CarFlag-Large		460000	1600×2048	322000/138000	138000	$(20 \sim 56) \times (85 \sim 265)$
AOLP	AC	2049	240×352	1368/681	681	$(25 \sim 32) \times (70 \sim 87)$
	LE		480×640	1292/757	757	$(28 \sim 48) \times (80 \sim 133)$
	RP		240×320	1438/611	611	$(30 \sim 58) \times (70 \sim 120)$
Caltech-cars		126	592×896	1626/126	126	$(23 \sim 59) \times (70 \sim 87)$
PKUData	G1	810	728×1082	322000/810	810	$(35 \sim 57) \times (145 \sim 184)$
	G2	700	728×1082	322000/700	700	$(30 \sim 62) \times (160 \sim 184)$
	G3	743	728×1082	322000/743	743	$(29 \sim 53) \times (145 \sim 184)$
	G4	572	1236×1600	322000/572	572	$(30 \sim 58) \times (158 \sim 170)$
	G5	1152	1200×1600	322000/1152	1438	$(20 \sim 60) \times (136 \sim 168)$

TABLE II
ABLATION EXPERIMENTS TO VALIDATE THE EFFECTIVENESS OF THE PROPOSED NETWORK. (a) AFFECT OF FEATURE MAP SIZE AFTER RoI POOLING ON PERFORMANCE. THE RESULT SHOWS THAT THE PERFORMANCE CAN BE IMPROVED BY A DENSER POOLING. HOWEVER, THE MODEL SIZE INCREASES ACCORDINGLY ("M" MEANS MILLION). BALANCING THE MODEL SIZE AND PERFORMANCE, WE CHOOSE 4×28 IN THE FOLLOWING EXPERIMENTS. (b) INFLUENCE OF FEATURE EXTRACTION METHOD IN PLATE RECOGNITION NETWORK ON PERFORMANCE. THE ADDITIONAL 2 CONVOLUTIONAL LAYERS PLUS RECTANGULAR MAX POOLING LEAD TO A BETTER PERFORMANCE

(a)

Size	End-to-end Performance (%)	Model size in detection net
4×16	95.45	$73M$
4×20	96.13	$90M$
4×24	96.76	$107M$
4×28	**97.13**	$124M$
4×32	97.15	$141M$

(b)

Method	End-to-end Performance (%)
Convs.+pooling	**97.13**
Average pooling	91.83
Max pooling	95.61

training images, data augmentation is implemented by rotation and affine transformation.

The third dataset is Caltech-cars (Real) 1999 dataset [37], which has license plates from America. It consists of 126 images with resolution of 592×896. To test the effectiveness of our framework on this dataset, we collect extra 1626 images with America license plates from [38], [39] to train the end-to-end model. Data augmentation is implemented by translation and rescaling.

The fourth dataset is issued by Yuan *et al.* [13], and denoted as "PKUData". It has 3977 images with Chinese license plates captured from various scenes. This dataset is also used only for test, by applying the well-trained model from CarFlag-Large. It is categorized into 5 groups (*i.e.,* G1-G5) corresponding to different configurations, as introduced in [13]. As there are only the plate bounding boxes given in the ground-truth file, we merely evaluate the detection performance on this dataset.

We summarize related information of all these datasets in Table I for reference.

B. Evaluation Criterion

To evaluate the "End-to-end" performance with both detection and recognition results considered, we follow the "End-to-end" evaluation protocol for general text spotting in natural scene [40] as they have similar application scenario. Define IoU as

$$\text{IoU} = \frac{\text{area}(R_{\text{det}} \cap R_{\text{gt}})}{\text{area}(R_{\text{det}} \cup R_{\text{gt}})} \tag{6}$$

TABLE III
EXPERIMENTAL RESULTS ON CARFLAG-LARGE DATASET. WE COMPARE BOTH PERFORMANCE AND RUNNING SPEED OF OUR JOINTLY TRAINED NETWORK WITH A TWO-STAGE BASELINE METHOD. THE JOINTLY TRAINED NETWORK ACHIEVES NOT ONLY HIGHER ACCURACIES ON BOTH DETECTION AND "END-TO-END" PERFORMANCE, BUT ALSO IN A SHORTER TIME

Method	End-to-end Performance (%)	Detection-only Performance (%)	End-to-end Speed (per image single scale) (ms)
Ours(Jointly-trained)	**97.13**	**98.33**	**310**
Ours(Two-stage)	94.09	97.05	450

where R_{det} and R_{gt} are regions of the detected bounding box and the ground-truth respectively.

The bounding box is considered to be correct if its IoU with a ground-truth bounding box is more than 50% (IoU > 0.5), and the plate labels match, *i.e.,* all characters in the plate are correctly recognized. It should be note that we denote all Chinese characters in license plates as '*', since the training images in CarFlag-Large are all from one province and use the same Chinese character. The trained network can not be used to distinguish other Chinese characters. F-measures are calculated and presented in our experiments which synthesize both precision and recall according to the following equation:

$$\text{F-measure} = \frac{2 \times (\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \tag{7}$$

TABLE IV

EXPERIMENTAL RESULTS ON AOLP DATASET. AC (ACCESS CONTROL) IS THE EASIEST DATASET WHERE IMAGES ARE CAPTURED WHEN VEHICLES PASS A FIXED PASSAGE WITH A LOWER SPEED OR FULL STOP. LE (LAW ENFORCEMENT) DATASET CONSISTS OF IMAGES CAPTURED BY ROADSIDE CAMERA WHEN A VEHICLE VIOLATES TRAFFIC LAWS. RP (ROAD PATROL) REFERS TO THE CASES THAT THE CAMERA IS HELD ON A PATROLLING VEHICLE, AND THE IMAGES ARE TAKEN WITH ARBITRARY VIEWPOINTS AND DISTANCES. WE COMPARE OUR PROPOSED METHOD WITH OTHER STATE-OF-THE-ART METHODS ON BOTH PERFORMANCE AND RUNNING SPEED. OUR JOINTLY-TRAINED NETWORK SHOWS IMPROVED PERFORMANCE FOR IMAGES WITH LICENSE PLATES IN NEARLY HORIZONTAL POSITION

Method	End-to-end Performance (%)			Detection-only Performance (%)			End-to-end Speed (per image single scale) (ms)
	AC	LE	RP	AC	LE	RP	
Hsu *et al.* [12]	–	–	–	96	95	94	**260**
Li *et al.* [24]	94.85	94.19	**88.38**	98.38	97.62	95.58	$1000-2000$
Ours(Jointly-trained)	**95.59**	**96.43**	83.80	**99.12**	**99.08**	**98.20**	400

As to the detection-only performance, we follow the criterion used in [13] for fair competition, *i.e.,* a detection is considered to be correct if the license plate is totally encompassed by the bounding box, and IoU > 0.5.

C. Performance Evaluation on CarFlag-Large

In this section, we run a number of ablations to analyze the network structure. A comparison experiment between our unified framework and a commonly used two-stage approach is also carried out to demonstrate the superiority of our end-to-end jointly trained network.

1) Network Structure Analysis: **Feature size after RoI pooling:** In order to determine the feature size X and Y after RoI pooling, a set of experiments are performed with different feature width Y. Experimental results in Table II(a) show that a longer Y will result in a better performance, but accompany with more parameters. By considering both the performance and the model size, we choose $X = 4$ and $Y = 28$ in the following experiments.

Feature extraction method in plate recognition network: As we illustrated in section III-A.5, two additional convolutional layers with ReLUs and rectangular poolings are added at the beginning of the plate recognition network, which convert the region feature map **Q** into a feature vector **V** for BRNNs processing. In Table II(b), we compare this to using solely average pooling or max pooling. 2-D average or max pooling with the kernel size as 4×4 and the step size as 1×1 is adopted before BRNNs respectively. Experimental results in Table II(b) suggest that the additional CNN layers can learn more discriminative features and lead to a better end-to-end performance.

2) Unified vs. Stepwise Framework: As illustrated in Figure 5, a commonly used two-stage approach implements plate detection and recognition by two separated models. Plate detection is carried out firstly. The detected objects are cropped out and then recognized by another different model. In contrast, our proposed network outputs both detection and recognition results at the same time, with a single forward pass and requiring no image cropping. The convolutional features are shared by both detection and recognition networks, which omits feature re-computation. For simplicity, we denote our jointly trained network as "Ours (Jointly-trained)", and the two stage approach as "Ours (Two-stage)". The model used only for plate detection is denoted as "Ours (Detection-only)".

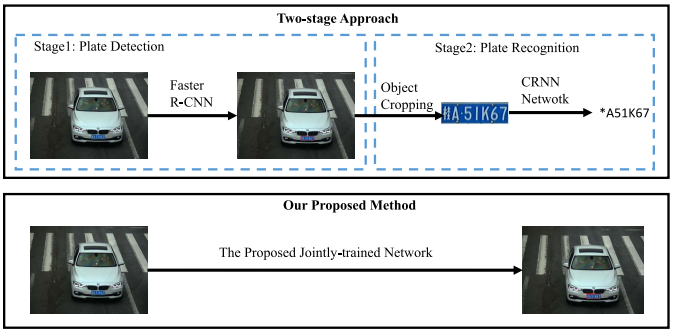

Fig. 5. Two-stage approach VS. our proposed method. In the two-stage approach, after license plate detection by Faster R-CNN, we crop the detected license plates from the image, and then recognize them by another separate model (CRNN in this paper). The features need to be re-computed during recognition phase. In contrast our proposed network takes an image as input, and produces license plate bounding boxes and plate labels in one-shot. It avoids some intermediate processes like image cropping, and share computation for convolutional feature extraction.

For fair competition, we train a Faster R-CNN [32] model using the 322000 training images for plate detection only. We modify the scales and shapes of anchors as the ones we used in this paper so that they fit the license plates. The network is also trained with $200K$ iterations, using the same initial parameters and learning rates. As to the plate recognition, we employ CRNN framework [27], which produces the state-of-the-art performance on general text recognition. It is an end-to-end framework for cropped word recognition, including CNN layers, RNN layers and CTC for transcription, from bottom to top. We crop the groud-truth license plates from the 322000 training images, and resize them to 160×32 pixels. Then we fine-tune the CRNN model with these training data.

In order to boost the performance, we rescale the input image into multiple sizes during test phase for both our proposed network and the detection-only Faster R-CNN network. The input images are resized to the shorter side of 600, 1200 pixels respectively, while the longer side less than 1500 pixels. With our framework, both detection and recognition results come out together, while with the two-stage approach, we crop the detected bounding boxes from input images, resize them to 160×32 pixels, and then feed into the trained CRNN model for recognition. Only bounding boxes with classification score larger than 0.95 are kept and merged via NMS. Considering that there is only one plate labeled as ground-truth per image, we finally choose the one that has 7

Fig. 6. Example results for open wide car license plate detection and recognition by our jointly trained model. Images in the first line are from CarFlag-Large, the second line are from AOLP, the third line are from PKUData and the fourth line are from Caltech-cars. The results demonstrate that our model can detect and recognize car license plates under various photographing conditions, such as day and night, sunny and rainy days, *etc*.

characters recognized and/or with the highest detection score for evaluation. The test results are presented in Table III. Our jointly trained network gives the "End-to-end" performance with F-measure of 97.13% on 138000 test images. It is around 3% higher than the results by the two-stage approach, which demonstrates the advantage of end-to-end training for both detection and recognition in a unified network. The learned features are more informative, and the two subtasks can help with each other.

In terms of the computational speed, the unified framework takes about 310*ms* per image for a forward evaluation on the single small input scale, while the two-stage approach needs around 450*ms* to get both detection and recognition results, as it needs to implement image cropping and CNN feature re-calculation.

We also compare the detection-only performance. Our jointly trained network produces a detection accuracy of 98.33%, which is 1% higher than the result given by detection-only Faster R-CNN network. This result illustrates that car license plate detection can be improved with the multi-task loss used during training time.

Some experimental results using our jointly trained network are presented in the first row of Figure 6, which show that our model can deal with images under different illumination conditions.

TABLE V

EXPERIMENTAL RESULTS ON CALTECH-CARS DATASET. OUR METHOD ACHIEVES THE BEST DETECTION PERFORMANCE COMPARED TO THE PREVIOUS METHODS

Method	Detection-only Performance (%)	End-to-end Performance (%)
Zhou *et al.* [10]	89.83	-
Tian *et al.* [42]	90.70	-
Li *et al.* [24]	96.38	-
Kim *et al.* [43]	97.60	-
Ours(Jointly-trained)	**98.04**	**94.12**

D. Performance Evaluation on AOLP

In this section, we compare the "End-to-end" performance of our method with other state-of-the-art methods on AOLP dataset, which has license plates from Taiwan. Note that the network is only trained with 15*K* iterations because of the small number of training images in this dataset. Moreover, since the sizes of license plates in AOLP are almost the same, and the ratios between license plates and images sizes are also similar. For this dataset, we only use a single image scale with shorter side as 700 pixels in test phase.

The detection and recognition results are presented on the second row in Figure 6. Comparison results with other methods in Table IV show that our approach performs better on AC and LE subsets with the "End-to-end" evaluation criterion.

TABLE VI

EXPERIMENTAL RESULTS ON PKUDATA. DETECTION PERFORMANCE AND RUNNING SPEED ARE COMPARED BETWEEN OUR PROPOSED METHOD AND OTHER STATE-OF-THE-ART METHODS. G1 - G5 CORRESPOND TO DIFFERENT IMAGE CAPTURING CONDITIONS. OUR JOINTLY TRAINED NETWORK ACHIEVES A AVERAGE DETECTION RATIO OF 99.73%, WHICH IS 2% HIGHER THAN THE PREVIOUS BEST PERFORMANCE. IN ADDITION, THE JOINTLY TRAINED NETWORK, WHICH INTEGRATES BOTH DETECTION AND RECOGNITION LOSSES, PERFORMS BETTER THAN THAT TRAINED ONLY WITH THE DETECTION INFORMATION

Method	Detection Performance (%)						Detection Speed (per image single scale) (ms)	End-to-end Speed (per image single scale) (ms)
	G1	G2	G3	G4	G5	Average		
Zhou *et al.* [10]	95.43	97.85	94.21	81.23	82.37	90.22	475	-
Li *et al.* [18]	98.89	98.42	95.83	81.17	83.31	91.52	672	-
Yuan *et al.* [13]	98.76	98.42	97.72	96.23	97.32	97.69	**42**	-
Ours(Detection-only)	**99.88**	99.71	99.46	99.83	98.68	99.51	283	-
Ours(Jointly-trained)	**99.88**	**99.86**	**99.60**	**100**	**99.31**	**99.73**	279	310

It also gives the best performance for plate detection on all three subsets, with averagely 2% higher than the sliding window based method used in Li *et al.* [24], and 4% higher than the edge based method used in Hsu *et al.* [12]. As to the computational speed, out network takes about $400ms$ to get both detection and recognition results, while Li *et al.*'s method [24] costs $2-3s$, and Hsu *et al.*'s [12] approach needs averagely $260ms$.

It should be noted that in Table IV, the "End-to-end" performance on RP subset is worse than that in [24]. That may be because the license plates in RP have a large degree of rotation and projective orientation. In [24], the detected license plates are cropped out and Hough transform is employed to correct the orientation. In contrast, our method does not explicitly handle the rotated plates. Integrating spatial transform network into our end-to-end framework may be a solution, referring to [41], which is a future work.

E. Performance Evaluation on Caltech-Cars

To further evaluate the effectiveness of the proposed framework, in this section, we test the "End-to-end" performance of our method on the Caltech-cars dataset [37], which has license plates from America. We fine-tune the model trained by CarFlag-Large dataset with about 1626 training images and test the model on Caltech-cars. Experimental results in Table V show that our framework is also feasible for detecting and recognizing license plates from America. It achieves the best detection performance compared to the previous methods. It should be note that this dataset does not include any training data. Although we collect 1626 images with America car license plates, they are not very consistent with each other. We believe the end-to-end performance can be further improved if we have enough consistent training data.

F. Performance Evaluation on PKUData

Because the ground-truth file in PKUData only provides the plate bounding boxes, we simply evaluate the detection performance on this dataset. Both the detection accuracy and computational efficiency are compared with other methods [10], [13], [18]. We use the same model trained by the CarFlag-Large dataset, as they are both datasets with Chinese license plates.

Images on the third line of Figure 6 show examples with both detection and recognition results. The detection-only results by our approach and other three methods are presented in Table VI. Our jointly trained model demonstrates absolute advantage on all 5 subsets, especially on G4, where we achieve 100% detection rate. This result proves the robustness of our approach in face of various scenes and diverse conditions. Qualitatively, our jointly trained network achieves a average detection ratio of 99.73%, which is 2% higher than the previous best performance.

In addition, the detection performance by our jointly trained network is slightly better than that by the detection-only network as seen from Table VI. This is consistent with the outcome on CarFlag-Large dataset, and proves again that the detection performance can be boosted when training with the label information.

In terms of computational speed, Yuan *et al.*'s [13] method is relatively faster than ours', since they use simple linear SVMs, while we use deep CNNs and RNNs.

V. CONCLUSION

In this paper we have presented a jointly trained network for simultaneous car license plate detection and recognition. With this network, car license plates can be detected and recognized all at once in a single forward pass, with both high accuracy and efficiency. By sharing convolutional features with both detection and recognition network, the model size decreases largely. The whole network can be trained approximately end-to-end, without intermediate processing like image cropping or character separation. Comprehensive evaluation and comparison on three datasets with different approaches validate the advantage of our method. In the future, we will extend our network to multi-oriented car license plates. In addition, with the time analysis, it is found that NMS takes about half of the whole processing time. Hence, we will optimize NMS to accelerate the processing speed.

REFERENCES

[1] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. Asian Conf. Comput. Vis.*, 2011, pp. 770–783.

[2] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3538–3545.

[3] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1457–1464.

[4] T. Wang, D. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Nov. 2012, pp. 3304–3308.

[5] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "PhotoOCR: Reading text in uncontrolled conditions," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 785–792.

[6] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 512–528.

[7] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, 2016.

[8] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A fast text detector with a single deep neural network," in *Proc. Nat. Conf. Artif. Intell.*, 2017, pp. 4161–4167.

[9] 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.

[10] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4269–4279, Sep. 2012.

[11] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 377–392, Sep. 2006.

[12] G.-S. Hsu, J.-C. Chen, and Y.-Z. Chung, "Application-oriented license plate recognition," *IEEE Trans. Veh. Technol.*, vol. 62, no. 2, pp. 552–561, Feb. 2013.

[13] 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.

[14] 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.

[15] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, pp. 42–53, Mar. 2004.

[16] S. Yu, B. Li, Q. Zhang, C. Liu, and M.-Q. H. Meng, "A novel license plate location method based on wavelet transform and EMD analysis," *Pattern Recognit.*, vol. 48, no. 1, pp. 114–125, 2015.

[17] I. Giannoukos, C.-N. Anagnostopoulos, V. Loumos, and E. Kayafas, "Operator context scanning to support high segmentation rates for real time license plate recognition," *Pattern Recognit.*, vol. 43, no. 11, pp. 3866–3878, 2010.

[18] 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.

[19] D. F. Llorca *et al.*, "Two-camera based accurate vehicle speed measurement using average speed at a fixed point," in *Proc. 19th Int. Conf. Intell. Transp. Syst.*, Nov. 2016, pp. 2533–2538.

[20] 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.

[21] Y. Hou, X. Qin, X. Zhou, X. Zhou, and T. Zhang, "License plate character segmentation based on stroke width transform," in *Proc. 8th Int. Congr. Image Signal Process.*, Oct. 2015, pp. 954–958.

[22] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–13.

[23] 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.

[24] 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

[25] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.

[26] L. Neumann and J. Matas, "On combining multiple segmentations in scene text recognition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 523–527.

[27] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[28] M. Busta, L. Neumann, and J. Matas, "Deep TextSpotter: An end-to-end trainable scene text localization and recognition framework," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2223–2231.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, Sep. 2014.

[30] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[31] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern. Recognit.*, Jul. 2017, pp. 6517–6525. [Online]. Available: https://arxiv.org/abs/1612.08242

[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[33] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, "DeepText: A unified framework for text proposal generation and text detection in natural images," *CoRR*, vol. abs/1605.07314, May 2016.

[34] R. Girshick, "Faster R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, Dec. 2014.

[37] (2003). *Caltech Plate Dataset*. [Online]. Available: http://www.vision.caltech.edu/html-files/archive.html

[38] L. Dlagnekov, "Video-based car surveillance: License plate, make, and model reconition," M.S. thesis, Dept. Comput. Sci., Univ. California, San Diego, CA, USA, 2015.

[39] (2013). *Benchmarks*. [Online]. Available: https://github.com/openalpr/benchmarks

[40] D. Karatzas *et al.*, "ICDAR 2015 robust reading competition," in *Proc. Int. Conf. Document Anal. Recognit.*, Aug. 2015, pp. 1156–1160.

[41] B. Shi, X. Wang, P. Lyv, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4168–4176.

[42] 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.

[43] S. Kim, H. Jeon, and H. Koo, "Deep-learning-based license plate detection method using vehicle region extraction," *Electron. Lett.*, vol. 53, no. 15, pp. 1034–1036, 2017.

Hui Li is currently pursuing the Ph.D. degree with the School of Computer Science, The University of Adelaide, Australia. Her research interests include text detection and recognition and deep learning.

Peng Wang received the bachelor's degree in electrical engineering and automation and the Ph.D. degree in control science and engineering from Beihang University, China, in 2004 and 2011, respectively. He was with the School of Computer Science, The University of Adelaide, for about four years. He is currently a Professor with the School of Computer Science, Northwestern Polytechnical University, China. His research interests are computer vision, machine learning, and artificial intelligence.

Chunhua Shen received the Ph.D. degree from The University of Adelaide. He was with the Canberra Research Laboratory, Computer Vision Program, National ICT Australia, for about six years. He was with Nanjing University, Nanjing, China, and The Australian National University, Canberra, ACT, Australia. From 2012 to 2016, he held an Australian Research Council Future Fellowship. He is currently a Professor with the School of Computer Science, The University of Adelaide. His research interests are in the intersection of computer vision and statistical machine learning.