2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

An End-to-End Neural Network for Multi-line License Plate Recognition

Yu Cao, Huiyuan Fu*, Huadong Ma
Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia,
Beijing University of Posts and Telecommunications, Beijing 100876, China
Email: yalecyu@gmail.com, {fhy, mhd}@bupt.edu.cn

***Abstract*—Currently, license plate recognition plays an important role in numerous applications and a number of technologies have been proposed. However, most of them can only work with single-line license plates. In the practical application scenarios, there are also existing many multi-line license plates. The traditional approaches need to segment the original input images for double-line license plates. This is a very difficult problem in the complex scenes. In order to solve this problem, we propose an end-to-end neural network for both single-line and double-line license plate recognition. It is segmentation-free for the original input license plate images. We view each of these whole images as a unit on feature maps after deep convolution neural network directly. A large number of experiments show that our method is effective. It is better than the state-of-the-art algorithms in SYSU-ITS license plate library data.**

Keywords*—multi-line; license plate recognition; character recognition; end-to-end neural network;

I. INTRODUCTION

Automatic license plate recognition plays an important role in numerous applications such as unattended parking lots, security control of restricted areas, traffic law enforcement, congestion pricing, and automatic toll collection, etc. Due to the different attributes and functions of vehicles, the laws and regulations of different countries have different rules on the license plate management. In Chinese traffic laws and regulations, single-line license plates are used in personal cars, police cars, military vehicles, embassy cars, trucks, buses, etc., and the double-line license plates are used in buses, passenger cars, low speed cars, trucks, armed police cars, motorcycles, etc. A large number of double-line license plates exist in the actual traffic scenes. License plate recognition is a multi-line text recognition task.

Most of the current license plate recognition methods only pay attention to the single-line license plate recognition task [1,2,4,5,6]. They identify the license plates by getting each character and then classifying the characters, or by end-to-end neural network. These methods can work well on single-line license plates. However, only few methods consider the double-line license plates [2,7,8]. These methods need to segment the input double-line license plates. Specifically, they require that each character of the plates should be segmented and recognized correctly. Each character segmentation and recognition precision will influence the final result. Moreover, they will face with a lot of workload obviously. It is urgent for an end-to-end way for the task.

Fig.1. Examples of multi-line license plates. It contains single-line and double-line license plates.

To the best of our knowledge, it is maybe the first time to propose an end-to-end neural network for multi-line license plate recognition. The main contribution of this paper is a novel neural network model, whose network architecture is specifically designed for recognizing the multi-line license plates. We work each of these whole images as a unit on feature maps after deep convolution neural network [3] directly. Then, the obtained feature maps are reorganized. Each feature map is represented as a sequence-like feature map for all single-line and double-line license plates. The bi-directional long short-term memory [14] is used on the obtained feature maps to calculate the sequence labels. By assembling these sequence label information, we can get the final license plate recognition results. The proposed neural network can be applied into many of the license plate recognition based systems due to that it can be trained and tested very conveniently. Extensive experimental results demonstrate that our proposed method achieves significant improvement comparing to current state-of-the-art approaches.

The rest of this paper is organized as follows: Section 2 introduces some approaches relevant to our work. In Section 3, the proposed approach is presented in detail. Experimental results are described in Section 4. Finally, we conclude the work in Section 5.

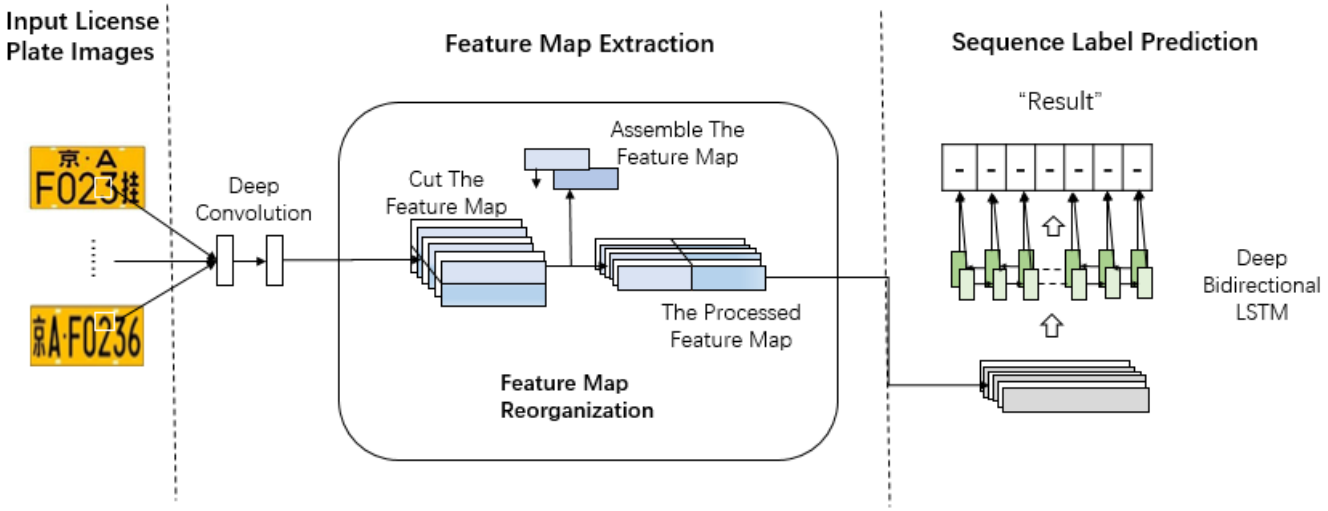

Fig.2. Our proposed end-to-end neural network for multi-line license plate recognition. The single-line images can be considered as a special subset of double-line images. For single-line and double-line character images, the convolution neural network extracts the feature values and summarizes the results as a $(1 \times C \times 2 \times width)$ feature maps. Where 1 is the batchsize and c is channels. The height and width of the feature maps will change via reorganizing. The Bi- LSTM processes the sequence-like feature map and predicts the final sequence labels.

II. Related Work

Many recent successful license plate recognition systems are developed based on deep learning. License plate recognition includes single-line and double-line identification.

For single-line license plate recognition, they always split characters by binarization and normalization [2], then classify them by machine learning. Or they use the deep neural network directly to recognize the characters. Deep recurrent neural networks have achieved great success in speech recognition [4]. The text recognition has the characteristics of timing just like that the speech recognition and the depth recurrent neural network is explored in end-to-end text recognition applications [5]. An end-to-end trainable neural network for image-based sequence recognition has been proposed and it has been applied to scene text recognition by Shi, Bai, and Yao et al [6]. It's just a combination of CNN and RNN. After computing feature maps by CNN, long and short memory can identify the text in the image according to process images by timing information.

For multi-line license plate recognition, the mature methods include preprocessing with binarization which results in segment [2] and position the text box by positioning algorithm, then they can extract single-line text information to identify. The text characters are located by sliding window [7] and non-maximal value method, or through the OpenCV [8] method, they can get the text box. After positioning the text box and extraction of multi-line license plate, the last step is to identify every line by the single line method. Since the entire task is segmented, the entire architecture can't be end-to-end with training and testing. The various aspects of these architectures interact with each other, thereby increasing the workload and complexity of the entire project.

III. Our Approach

In this section, we will introduce the framework of the proposed method. The Fig. 2 shows the network architecture. It includes the deep convolutional network, feature map reorganization and deep recurrent neural network part. Based on these basic parts, we describe how to use the proposed network to solve the challenges of multi-line license plate recognition. We will explain some of the key parts of the network. First of all, features are extracted by deep convolutional neural network. Then the feature map is reorganized. Finally, the Bi-LSTM predicts the sequence feature maps [11].

A. Feature Map Extraction

Feature extraction: Our approach will be more focused on the operation of the feature map, different from the previous manual methods in the original implementation. Through the training of the objective function, the deep neural network can calculate the features of the image and iterate towards our goal. There are already well-established experiments and studies in the feature map, such as the Faster R-CNN [9] network. The operation of the sliding window is used in the feature map, but unlike the conventional method of using the sliding window in the original image, the efficiency and effect are improved.

Convolutional neural networks consist of multiple convolutional layers, rule layers, and pooling layers, similar to the standard convolutional neural network. Such component is used to extract a sequential feature representation from an input image. By convoluting the neural network, all images are scaled to a certain height and width. Specifically, each feature vector of a feature sequence is generated from left to right on the feature maps by column. For the task of double-line character recognition, we expect convolution neural network to calculate the image features, and make the feature map keep

the logical position of the original image. The original image is scaled to a smaller feature map, which is combined by the useful information extracted from the original images. As the layers of convolution, max-pooling and elementwise activation function operate on local regions, they are translation invariant. Therefore, each part of the feature map corresponds to a certain area of the original image, and the relative position of the feature map has a strong relationship with the relative position of the original image.

Receptive field: In the convolutional neural network, the size of the input layer that determines an element in the output of a layer is called the receptive field [10]. For the task of multi-line license plate recognition, we need to pay attention to the receptive fields of convolutional neural networks to deal with the extreme conditions. The receptive field is determined by the size of kernel and the size of stride. For example, for a kernel of size 2, the step of the convolution kernel is 1. Each element in the output is the maximum value in the region of 2x2 that corresponds to the input, so the receptive field size for this layer is 2x2. The output value of one element corresponds to the input result of the calculation of the four elements. The receptive fields are present in convolutional neural networks, convolution layers and pooling layers. The formula for calculating the receptive field is:

$$\begin{aligned} r_1 &= 1 \\ r_n &= f(r_{n-1}, s, k) = (r_{n-1} - 1) * s + k \end{aligned} \tag{1}$$

where r_0 is the size of the first perception wild, r_n is the receptive field of this layer, s is the step of the kernel, k is the size of the kernel. Through the calculation of the formula, we can get the receptive field of this layer. The calculation of the receptive field has three explanations as following:

- The size of the receptive field of the output characteristic pixel of the first layer is equal to the size of the filter.
- The receptive field's size of the deep convolutional layer and the pooling layer is related to the filter size and step size of the previous neural layer.
- When we calculate the size of the receptive field, you can ignore the edges of the image, regardless of the size of the padding.

The following figure shows an example of a manually-designed convolution kernel for feature calculation, relative position invariance, and receptive fields. Artificial convolution kernel is only a description of convolutional neural network. Training through a large amount of data, the higher the information calculated by the convolution kernel, the higher the level of the output feature map, the stronger the ability of the model to process the task. Many papers have characterized the ability of convolutional neural networks in image processing. In experiments and tasks of these paper, convolution neural network has an excellent effect.

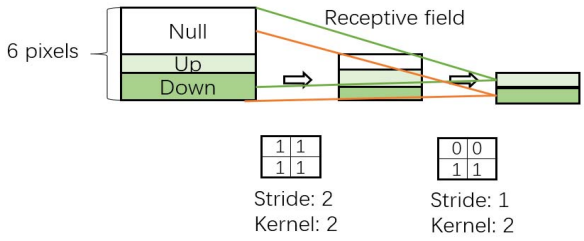

Fig.3. "Null" does not contain the characteristic information, "Up" represents the character information of the upper row, and "Down" represents the character information of the lower row.Our convolution kernel by calculating the retention of the upper row of text information and the next row of text information, remove the empty information. To reduce the size of the image, the final height is 2, half the width of the original image(Ignore the pad, bias, etc.). The final calculated height of 2 characteristics of the map, the upper and lower part of the receptive field in the original image height of 5 pixels receptive field.

B. Feature Expression

Feature map reorganization: We give the goal of convolution neural network, through neural network computing, feature extraction and function scaling. We reorganization the feature map by cutting the feature map and connecting the feature map. This step is a steady method and is not disturbed by other parts. For double-line license plates, the feature map is finally calculated and scaled to the height of 2 features. By designing deep neural networks, we ensure that the network has enough receptive field to avoid uncontrollable extremes. We reconstruct the feature maps on the final computed feature map through deep neural networks, resulting in a height of 1 width. The processed feature maps have sequence features, left to right timing features.

$$\begin{cases} h_1 = 0 \\ h_i, h_{i+1} - 1 = (i-1) * a, i * a - 1 \\ w_i = w \end{cases} \tag{2}$$

$$\begin{cases} W_1 = 0 \\ W_i, W_{i+1} - 1 = (i-1) * a, i * a - 1 \\ H_i = h / a \end{cases} \tag{3}$$

$$H, W = h / a, w * a \tag{4}$$

where h_i is the height of the i-th input data. w_i is the width of the i-th input data. W_i is the width of the i-th ouput data. H_i is the height of the i-th output data. H and W are the new height and width after the operation, h and w are the dimensions of the feature map before the operation. According to experience, we set a equal to the maximum number of lines in the task.

The whole solution could be obtained from the **Algorithm** 1.

Algorithm 1 Feature map reorganization
Input: The feature map
Output: The reorganized feature map

```
if a != 0 or h % a == 0 then
    exit
end if
for each feature map do
    for i = 1, 2,…, a do
        Get the location of i-th input data block by Eq. 2
        Get the location of i-th output data block by Eq. 3
        Copy the input data to the output data
    end for
end for
```

Mix multiple lines: When dealing with the identification of single-line characters, the feature map does not make any changes. For the recognition of double-line characters, we hope to reduce the dimension by convolutional neural network to generate a characteristic map with height of 2 finally, and cut the characteristic map from the center line. For mixed multi-line license plates, feature maps are generated. The feature map of all images are restructured. We can regard the single-line license plate as the special case of the double-line license plate. An image with a small number of lines is one type of image having a large number of lines, so that it can be processed in multiple images of lines.

C. Sequence Label Prediction

Bi-LSTM: Establish a deep bidirectional Recurrent Neural Network after convolutional neural network [11], as the recurrent layers. The recurrent layers have the ability to process the feature map from left to right. By combining the contextual information, the processing results of the current frame are combined with the previous memory and forgetting results, the current information is processed for prediction, and the result is passed to the following frame for processing through the forgotten gate and the memory gate. In our task, character sequence recognition has a timing characteristic from left to right. For multi-line license plates, the entire feature map recognition also has timing characteristics through data reorganization

The problem with unidirectional RNNs is that only the information before t can be used when t is classified, but the information at future time may also need to be used when t is classified. The bi-directional RNN model tries to solve the problem. The bi-directional RNN maintains two hidden layers at any time t. One hidden layer is used for transmitting information from left to right and another hidden layer is used for from right to left information dissemination recorded as. Many articles and experiments show that bidirectional circular networks have better performance in areas such as speech recognition. Therefore, we use bi-directional LSTM networks to predict the timing after the convolution neural network.

CTC-Loss: We adopt the conditional probability defined in the Connectionist Temporal Classification (CTC) layer proposed by Graves et al. [12]. The probability is defined for label sequence l conditioned on the per-frame predictions y = y_1 …. y_T, and it ignores the position where each label in l is located. Consequently, when we use the negative log-likelihood of this probability as the objective to train the network, we only need images and their corresponding label sequences, avoiding the labor of labeling positions of individual characters

$$p(l \mid y) = \sum_{\pi : B(\pi)=1} p(\pi \mid \mathrm{y}) \tag{5}$$

Where the probability of π is define as $p(\pi \mid y) = \prod_{t=1}^{T} y_{\pi_t}^t$, $y_{\pi_t}^t$ is the probability of having label π_t at time stamp t.

D. Network Training

The input to the network is sequence tag information for images and images. Denote the training dataset by X = {I_i , L_i}, where Ii is the training image and li is the ground truth label sequence. The objective is to minimize the negative log-likelihood of conditional probability of ground truth:

$$\mathrm{O} = -\sum_{I_i, L_i \in X} \log p(I_i \mid \mathrm{y}_i) \tag{6}$$

where y_i is the sequence produced by the recurrent and convolutional layers from I_i. This objective function calculates a cost value directly from an image and its ground truth label sequence. Therefore, the network can be end-to-end trained on pairs of images and sequences, eliminating the procedure of manually labeling all individual components in training images.

The network is trained with stochastic gradient descent (SGD) [13]. Gradients are calculated by the back-propagation algorithm. In particular, in the transcription layer, error differentials are back-propagated with the forward-backward algorithm, as described in. In the recurrent layers, the Back-Propagation Through Time is applied to calculate the error differentials.

IV. EXPERIMENTS

A. Datasets

In order to verify the validity of the model, we tested its performance on a public dataset. We evaluate our method on the SYSU-ITS [14] Dataset. It includes a total of 1402 images, 958 single-line license plate images and 84 double-line license plate images. The license plate images in the SYSU-ITS functional evaluation image library are all selected from the HD images of road bayonets. Each image contains only one license plate, and the single character height of the license plate area in the image is greater than 25 pixels). Each license plate character imaging clear, no adhesion, light well, uniform, license plate image vertical tilt angle and horizontal tilt angle is small (negligible). **Fig.4** shows some of the samples in the data set.

License plate dataset includes 72 kinds of character recognition, 38 Chinese labels, 10 numeric labels and 24 English alphabets. The data set includes two styles of license plates, single-line license plates and double-line license plates. In order to compare the effects of the recognition, we do not consider license plates that are not correctly recalled by the detector. We do not discuss the identification of Chinese characters, because of the design of personal privacy issues. Define IoU as follows:

$$IoU = \frac{area(R_{\det} \cap R_{gt})}{area(R_{\det} \cup R_{gt})} \quad (7)$$

where $R_{\det}$ and R_{gt} are regions of the detected bounding box and ground-truth respectively. The bounding box is considered to be correct recalled if its IoU with a ground truth bounding box is more than 70% (IoU > 0.7).

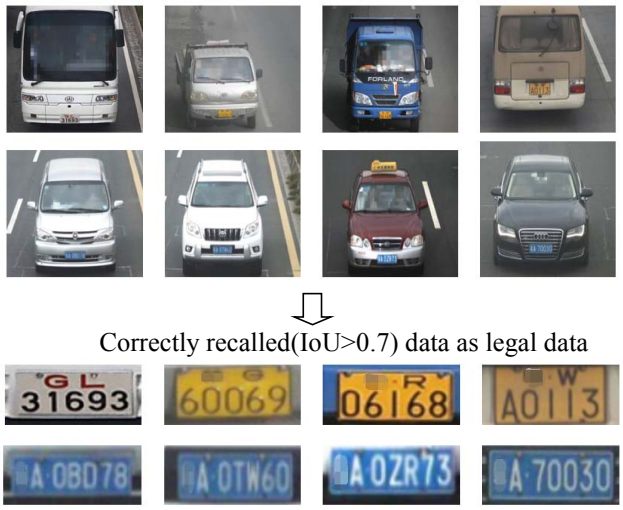

Fig.4. The data set contains single-line and double-line license plates. We do not discuss the identification of Chinese characters, because of the design of personal privacy issues

B. Network Structure

In the experiment, we use a specific network configuration shown in Table 1. The architecture convolution neural network is based on the VGG-VeryDeep architectures [15]. We adjust the network to handle the multi-line license plate recognition project. We need ensure that the neural network has enough receptive fields in the vertical dimension to handle the extreme cases of textual alignment.

The network has deep convolutional layers and recurrent layers. They are hard to train. The batch normalization [16] technique is extremely useful for training network of such depth. Two batch normalization layers are inserted after the Conv3 and conv5 layers respectively. With the batch normalization layers, the training process is greatly accelerated.

The experiment is based on the Caffe [17] framework with the basic configuration. This framework is to support the CNN and Bi-LSTM. And it need support the corresponding layer, such as batch normalization layer and ctc loss layer. The experiments are carried out on a workstation with an NVIDIA(R) Tesla(TM) M40 GPU.

Type	Network Structure		
	Shape	Configuration	Receptive field
Input	3x64x96		102
Conv1	64x64x96	K:3 S:1 P:1	100
Pool1	64x32x48	K:2 S:2	50
Conv2	128x32x48	K:3 S:1 P:1	48
Pool2	128x16x24	K:2 S:2	24
Conv3	256x16x24	K:3 S:1 P:1	22
BatchNorm	256x16x24		
Conv4	256x16x24	K:3 S:1 P:1	20
Pool3	256x8x12	K:2 S:2	10
Conv5	512x8x12	K:3 S:1 P:1	8
BatchNorm	512x8x12		
Conv6	512x8x12	K:3 S:1 P:1	6
Pool4	512x4x12	K:[2,1] S:[2,1]	3
Conv7	512x2x12	K:3 S:1 P:[0,1]	1
Reorganization	512x1x24		
Bi-LSTM	#hidden units:128		
Bi-LSTM	#hidden units:256		

Table.1. 'k', 's' and 'p' stand for kernel size, stride and padding size respectively. The effect of the edge of the image is ignored, when calculating the receptive field size.

C. Comparative Evaluation

The accuracy is the ratio of the number of correctly identified license plates to the number of correctly positioned license plates. We discuss the effectiveness of these methods by comparing the accuracy of these methods.

The method of Jaderberg [21] recognizes the license plate by locating each character and then classifying each character. It divides the entire recognition into two parts. It is a state-of-the-art method. It is validated on many datasets with good results. We use the correctly positioned license plate as the input data for this method.

We also compared our approach to other state-of-the-art methods. The methods include openly published methods and open interfaces [18,19,20]. We use the correctly positioned license plates as the input data for these methods. For public interfaces, we train it by using the same data set as our network. The method of Wang [19] recognizes the license plate by segmenting each character. The method of segmentation has a lot of human interference. We use the best result as a comparison. For other published methods, we train and code according to the parameters recommended in the paper. Fig.5 shows the comparison of the accuracy with other methods on single-line and double-line data. For single-line license plate recognition, our method and these methods are similar in accuracy. For double-line license plate recognition,

our method is much better than these methods. Table.2 shows the comparison of our method and other methods on multi-line license plate data. The experiments show that our method is easy to use and that our method is better than them in the multi-line license plate recognition.

Fig.6 shows the loss of training over time. Our method is trained for about five hours.

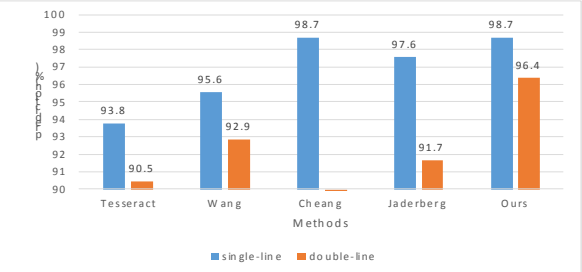

Fig.5. It shows the comparison of the accuracy with other state-of-the-art methods on single-line and double-line license plates data.

Method	Accuracy
Tesseract [18]	93.6
Wang [19]	95.4
Cheang [20]	94.3
Jaderberg [21]	97.1
Ours	**98.5**

Table.2. It shows the comparison of the accuracy with other state-of-the-art methods on multi-line license plates data.

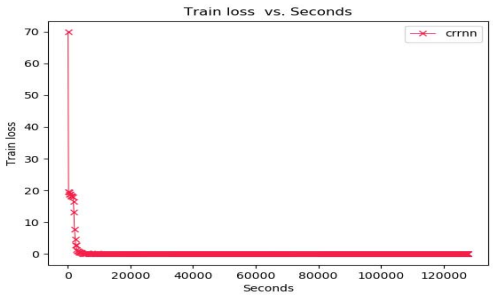

Fig.6. It shows the train loss over time.

V. CONCLUSIONS

In this paper, we propose an end-to-end neural network for multi-line license plate recognition. The experimental results demonstrate the effectiveness of the proposed method in comparison with several state-of-the-art algorithms.

ACKNOWLEDGMENT

This work was supported by the National Key R&D Program of China 2017YFB1003000, the NSFC-Guangdong Joint Fund under no. U1501254, the CCF-Tencent Open Fund, and the Funds for Creative Research Groups of China under no. 61421061.

REFERENCES

[1] Chang S L, Chen L S, Chung Y C, et al. Automatic license plate recognition[J]. IEEE transactions on intelligent transportation systems, 2004, 5(1): 42-53

[2] Gou C, Wang K, Yao Y, et al. Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(4): 1096-1107.

[3] Papandreou G, Chen L C, Murphy K, et al. Weakly-and semi-supervised learning of a DCNN for semantic image segmentation[J]. arXiv preprint

[4] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[C]//Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. IEEE, 2013: 6645-6649.

[5] Messina R, Louradour J. Segmentation-free handwritten Chinese text recognition with LSTM-RNN[C]//Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. IEEE, 2015: 171-175.

[6] Shi B, Bai X, Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(11): 2298-2304.

[7] Anagnostopoulos C N E, Anagnostopoulos I E, Loumos V, et al. A license plate-recognition algorithm for intelligent transportation system applications[J]. IEEE Transactions on Intelligent transportation systems, 2006, 7(3): 377-392.

[8] Liu X, Samarabandu J. An edge-based text region extraction algorithm for indoor mobile robot navigation[C]//Mechatronics and Automation, 2005 IEEE International Conference. IEEE, 2005, 2: 701-706.

[9] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[C]//Advances in neural information processing systems. 2015: 91-99.

[10] Olshausen B A, Field D J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images[J]. Nature, 1996, 381(6583): 607.

[11] Schuster M, Paliwal K K. Bidirectional recurrent neural networks[J]. IEEE Transactions on Signal Processing, 1997, 45(11): 2673-2681.

[12] Graves A, Gomez F. Connectionist temporal classification:labelling unsegmented sequence data with recurrent neural networks[C]// International Conference on Machine Learning. ACM, 2006:369-376.

[13] Bottou L. Large-scale machine learning with stochastic gradient descent[M]//Proceedings of COMPSTAT'2010. Physica-Verlag HD, 2010: 177-186.

[14] http://www.openits.cn/openData4/569.jhtml

[15] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[16] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//International Conference on Machine Learning. 2015: 448-456.

[17] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]//Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014: 675-678.

[18] https://github.com/tesseract-ocr/

[19] Wang Y, Ban X, Chen J, et al. License plate recognition based on SIFT feature[J]. Optik-International Journal for Light and Electron Optics, 2015, 126(21): 2895-2901.

[20] Cheang T K, Chong Y S, Tay Y H. Segmentation-free Vehicle License Plate Recognition using ConvNet-RNN[J]. arXiv preprint arXiv:1701.06439, 2017.

[21] Jaderberg M, Vedaldi A, Zisserman A. Deep Features for Text Spotting[C]// European Conference on Computer Vision. Springer, Cham, 2014:512-528.