


# Towards Human-Level License Plate Recognition

Jiafan Zhuang<sup>[0000-0003-3708-4634]</sup>, Saihui Hou<sup>[0000-0003-4689-2860]</sup>,  
Zilei Wang <sup>[0000-0003-1822-3731]</sup>, and Zheng-Jun Zha<sup>[0000-0003-2510-8993]</sup>

University of Science and Technology of China  
{jfzhuang, saihui}@mail.ustc.edu.cn, {zlwang, zhazj}@ustc.edu.cn

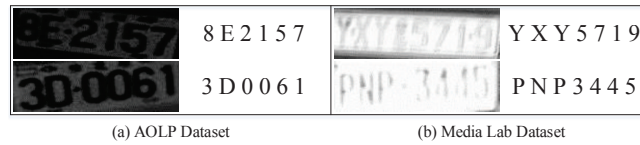
**Abstract.** License plate recognition (LPR) is a fundamental component of various intelligent transport systems, which is always expected to be accurate and efficient enough. In this paper, we propose a novel LPR framework consisting of semantic segmentation and character counting, towards achieving human-level performance. Benefiting from innovative structure, our method can recognize a whole license plate once rather than conducting character detection or sliding window followed by per-character recognition. Moreover, our method can achieve higher recognition accuracy due to more effectively exploiting global information and avoiding sensitive character detection, and is time-saving due to eliminating one-by-one character recognition. Finally, we experimentally verify the effectiveness of the proposed method on two public datasets (AOLP and Media Lab) and our License Plate Dataset. The results demonstrate our method significantly outperforms the previous state-of-the-art methods, and achieves the accuracies of more than 99% for almost all settings.

**Keywords:** License Plate Recognition (LPR), Semantic Segmentation, Convolutional Neural Networks (CNN), Character Counting

## 1 Introduction

License plate recognition (LPR) from images as a fundamental component is vitally important for various intelligent transport systems (ITS), such as security control of restricted areas [9, 28] and traffic safety enforcement [33, 16], as the license plate is generally used as the identification of a vehicle. So the LPR algorithm is always required to be accurate and efficient enough for facilitating the deployment and application in terminal equipments or transport systems. On the other hand, LPR in complex scenes that the vehicle images may be highly deformed or blurred is probably the main consideration in many real-world applications. For instance, the suspect vehicles are the mainly concerned targets that need to extract the structured information in the safety surveillance, while other normal ones can be completely ignored. But the acquired images for such vehicles are usually harder to analyze. Therefore, developing accurate and robust LPR is essential, especially for complex scenes.

LPR needs to handle the diverse illuminance conditions, image blurring, defacing, and vehicle deformation [10, 31, 2, 1, 26]. The existing works can be



**Fig. 1. Some example results recognized by our proposed method.** The images are from two public datasets: AOLP and Media Lab

roughly divided into two broad categories: traditional methods consisting of character detection and recognition, and recent sequential methods. The first type of methods are intuitive which first detect all characters in a license plate and then independently recognize the detected characters. In practice, the methods are too sensitive to the environmental factors and consequently error-prone [11]. Thus they are applicable only to the restricted scenarios with fixed conditions, e.g., parking entrances. The sequential methods [21, 32, 7, 6] are proposed in recent years, which first conduct sliding window to extract the sequential features of a license plate, and then recognize the license plate from the extracted features by a CNN or RNN model. Relatively, these methods can produce better recognition performance, but it is not enough to meet the requirements of real systems. Meanwhile, too many calls to deep learning models make these methods time-consuming, leading to low computational efficiency.

In this paper, we propose a novel LPR framework with aims of simultaneously improving the recognition accuracy and computational efficiency. It consists of two key components: semantic segmentation and character counting. Owing to such an innovative structure, our approach is robust to various image quality, and Fig.1 provides some example results of our method. To our knowledge, this work is the first attempt to adopt advanced semantic segmentation in LPR. Indeed, our proposed method can achieve outstanding recognition accuracy benefiting from the following facts. First, semantic segmentation directly processes an entire license plate image rather than subimages of individual characters, which helps to exploit more information from both global and local contexts. Second, we conduct the pixel-level recognition instead of character recognition in previous works, which can endure a small part of pixels to be misclassified without causing a failure of LPR. Third, our method does not require very precise character detection since semantic segmentation has completed the recognition function, but which is exactly one of the main obstacles in the traditional methods. On the other hand, our proposed method possesses higher computational efficiency because semantic segmentation once recognizes all pixels of an input image and only several times (far less than the number of characters in a license plate) of calls to the CNN model are needed in the subsequent counting procedure.

We experimentally evaluate the proposed method on two public license plate datasets (AOLP and Media Lab) and our own License Plate Dataset, which all are built using the images from real scenes and cover many difficult conditions. The results on public datasets consistently demonstrate that our method

is greatly superior to the previous state-of-the-art methods. And for most of settings, a recognition accuracy of more than 99% is achieved, which implies that our method is applicable to real-world systems, even for complex scenes. Furthermore, we also conduct error study by analyzing all failure samples produced by our method. It is observed that these samples are extremely difficult to recognize accurately, and most of them is very confusing even for human being. This indicates that the performance of our proposed method is very close to the human level.

## 2 Related work

Before the explosion of deep learning, almost all traditional LPR methods [11, 3] adopt a similar process flow, *i.e.*, character detection followed by separately character recognition. The differences among them mainly are minor implementation details. Here we briefly review the related works on character detection and recognition.

Character detection is to detect each character in a license plate image with a bounding box. Most of existing algorithms for character detection can be simply divided into two families. The first one is based on Connected Component Analysis (CCA) [2, 15]. After removing image noise and eroding the connecting area of neighboring characters in the binarized license plate image, CCA is performed to obtain all connected areas and each of them is regard as a character. The second family relies on the projection methods [13, 34, 12, 35, 18]. Different from the first one, they project the binary image horizontally to obtain the top and bottom boundary, and then vertically to segment each character.

Character recognition is to recognize each candidate extracted by character detection. Various methods have been proposed for this task. [13, 34, 15, 26] are based on the template matching that measures the similarity between the character and templates. These methods are computationally costly and easily affected by interference. The feature based methods use more discriminate features, *e.g.*, hand-crafted feature [35, 18], and a specific classifier, *e.g.*, probabilistic neural network (PNN) [2], HMM [12], and MLP neural network [27]. [15, 14] extract the LBP features and adopt a linear discriminant analysis (LDA) as classifier. With the arrival of deep learning, character recognition in the LPR task is usually implemented by a CNN network [25].

Recently, the advancement of deep learning techniques brings new ideas to find a solution without character detection. A typical method uses a sliding window way to extract sequential features of license plate, and then infers the character sequence by feeding the extracted features into a classification model. Here the models for feature extraction include sweep OCR [7], CNN [21], and FCN [32]. The inference models include HMM [7], RNN with LSTM [21], and customized NMS algorithm [32]. Benefiting from the elimination of character detection, these methods can achieve better recognition performance, which indeed are the state-of-the-art. However, these methods need to repeatedly apply the character classifier many times, leading to low computational efficiency. More-

over, the methods conduct the character recognition on a small patch of the license plate image, which is easily influenced by interference in complex scenes. A related work [29] utilizes single-histogram approach to perform pixel-wise prediction to identify the character regions, followed by the street name recognition with template matching.

In this work, we propose a novel LPR framework. Unlike the existing works, we adopt semantic segmentation to recognize a whole license plate followed by character counting. Our framework avoids the sensitive character detection and meanwhile utilizes global information of a license plate image. Besides, our method is computationally efficient without a sliding window profile.

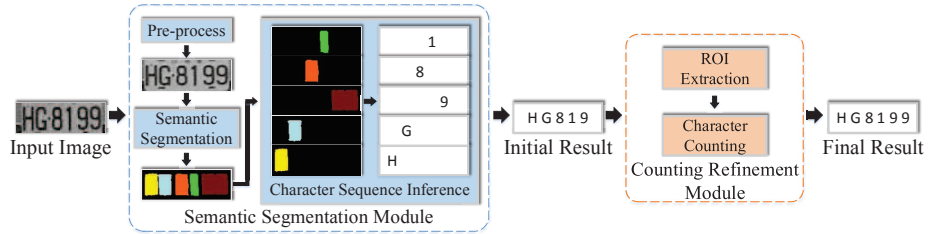
### 3 Our method

#### 3.1 Overview

Our purpose in this work is to invent an accurate and efficient license plate recognition (LPR) method by overcoming the drawbacks of previous works. To this end, we particularly introduce the semantic segmentation technique as the main component since it can directly process a whole license plate image. Here the input refers to the cropped subimages from raw images that only contain the license plate content with little background, which in practice is easily accomplished by applying some handy detection methods [13, 34, 12, 35, 18]. We illustrate the proposed LPR framework in Fig.2, which consists of two key modules: semantic segmentation and counting refinement.

Semantic segmentation is used to produce the semantic map of the input license plate image with the same size, whose value represents the character class of corresponding pixel. We then operate on the produced semantic maps to obtain the character sequence of a license plate. Compared with the character recognition in traditional methods, semantic segmentation can exploit more information in addition to the advantage of high computational efficiency, as the entire input image is fed once and both the global and local information from different network layers are fused. In this module, we conduct pre-processing on the input images and character sequence inference on the resulting maps besides semantic segmentation. Here the pre-processing is to make the images suitable for semantic segmentation, and only a simple projection is adopted in our implementation. As for the character sequence inference, an initial recognition result would be produced from the semantic map.

In the resulting semantic segmentation maps, we cannot distinguish the successive characters of the same type. To handle such situations, a counting refinement module is proposed to append to the semantic segmentation module in this work, which is used to predict the number of successive characters. In theory, the times to conduct the character counting for a license plate is at most half of the total number of characters. Thus this procedure is statistically time-saving, which together with semantic segmentation makes our method very efficient compared to the previous methods based on deep learning. Finally, we would obtain the character sequence of an input license plate.



**Fig. 2. Illustration of our proposed LPR framework.** The framework consists of two key modules: semantic segmentation and counting refinement. The semantic segmentation module is to produce the semantic map and initial character sequence, which includes pre-processing, semantic segmentation, and character sequence inference. The counting refinement module is to generate the final character sequence through counting characters. Best viewed in color

From Fig.2, we can observe the following features of the proposed LPR framework. First, we adopt the pixel-wise recognition to replace the character recognition in the traditional methods. Such replacement eliminates the character detection that is sensitive to imaging conditions and would deteriorate the overall performance of LPR, and meantime improves the robustness to image interference as the wrong prediction on a small proportion of pixels almost has no effect on the final results. Second, our framework processes a whole license plate once and avoids time-costly sliding window in the previous works, which thus is more efficient. Third, our proposed framework is flexible and upgradeable, in which each module can be assigned a different model. Thus one application can be equipped with the appropriate models by considering the actual demands in accuracy and efficiency. Next we will elaborate on the details of two key modules.

### 3.2 Semantic segmentation module

The semantic segmentation module is functionally to produce a semantic map and initial character sequence for an input license plate image. Here the semantic map has the same size as the input. In particular, we first pre-process the input image to make it more suitable for semantic segmentation, and then infer the initial character sequence by analyzing the semantic map. For pre-processing, we first find the character boundary by horizontally and vertically projection analysis, and then crop the main body of license plate image as the input. For semantic segmentation, we adopt a modified DeeplabV2 ResNet-101 model and put our focuses on the data preparation. For character sequence inference, we expect that the obtained initial character sequence are robust enough even when the segmentation results are not good with wrong predictions on some pixels.

**Segmentation groundtruth** In the LPR scenario, the results of semantic segmentation are essentially used to produce the character sequences, which is a



**Fig. 3. Illustration of box-level label for a license plate image.** The sample is from the AOLP dataset. Here different colors denote different character classes. The pixels not belonging to any character box are labeled as the background class (black). Best viewed in color

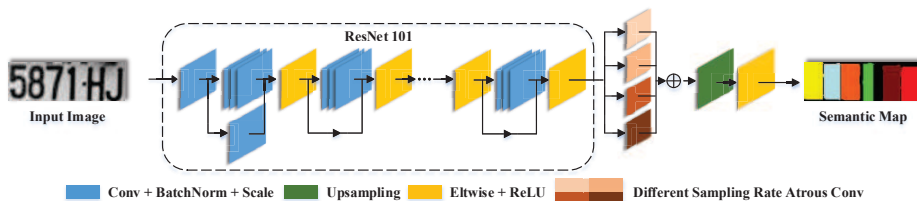
bit different from the usage in traditional segmentation tasks aiming to precisely predict the semantic label of each pixel. Thus we propose to adopt a relatively simple segmentation ground truth for each license plate in this work.

Considering the difficulty of labeling all pixels, we particularly use the wrapped bounding box of each character as its segmentation ground truth, instead of a subtle mask representing the character contour. All pixels in the bounding box are labeled the class of corresponding character, and other pixels not belonging to any character are labeled the background class. Fig.3 illustrates the ground truth generation of a license plate by an example. Obviously, the box-level annotation is more labor-saving than the pixel-level annotation, which benefits the proposed framework more operable than other semantic segmentation tasks.

Moreover, learning the bounding boxes of characters is more robust for LPR due to containing more pixels, unlike the prediction of character contours that is prone to be broken due to the thinness of labeled area. Consequently, some wrong prediction over a few pixels has almost no negative effect on the overall results in our method. On the other hand, the proposed segmentation groundtruth also makes the subsequent analysis of character sequence easier as a bigger semantic area has lower complication.

**Semantic segmentation model** Here we mainly consider the semantic segmentation models based on deep learning due to their achieved state-of-the-art performance. In theory, any advanced semantic segmentation model can be used in the proposed LPR framework, such as Fully Convolution Network (FCN) [24], DeepLabv2 [8], and SegNet [5]. In our implementation, a modified version of DeepLabv2 ResNet-101 is particularly adopted, as shown in Fig.4, whose performance has proven by evaluation to be good enough for LPR.

Here we simplify the original DeepLabv2 ResNet-101 model to boost the computational efficiency. The original model performs multi-scale process to obtain more accurate segmentation results, which essentially is to fuse the hierarchical global information. In the LPR task, however, the semantic areas of different characters have lower correlation. Meanwhile, our proposed analysis method is robust to wrong prediction on a few pixels. Thus in this work we only reserve the original resolution branch and remove other auxiliary branches. Actually, we conduct the comparative analysis between the simplified version and the original DeepLabv2 ResNet-101 by experiments. The results show that the modified one performs faster without involving an accuracy decrease on the final LPR results.



**Fig. 4. The modified DeepLabv2 ResNet-101 model.** The network mainly includes the ResNet-101 for feature extraction, atrous spatial pyramid pooling (ASPP), and upsampling layers. Best viewed in color

Specifically, with the fixed configuration (GeForce GTX 1080Ti), our version achieves 38 FPS, while the original DeepLabv2 only has 13 FPS.

**Model pre-training** Semantic Segmentation usually requires a massive collection of training samples to achieve accurate prediction. But the existing license plate datasets are commonly small in scale and cannot afford such data requirements. To address this issue, we propose to pre-train the model and then fine-tune it using the samples in a specific dataset. For pre-training, we use the synthetic images of license plate in this work since the data scale is extensible and the ground truth is easily obtained that can be concurrently generated in the synthesis process.

Here we propose to synthesize the license plate images by the combination of different characters. Specifically, we first set a base dataset containing many license plate images, in which various images with different illumination, color, blurring, and viewpoint need be embodied for each character. In our implementation, a subset of our own License Plate Dataset in Section 4.4 is particularly employed. Then we sample substantial character images by cropping from the license plate images, which form the materials of data synthesis. The cropping operation is natural since we use a bounding box to annotate each character. Finally, we synthesize plenty of license plate images by replacing one character region in a license plate image with another character image. To generate balanced samples, we randomly choose the base license plate image and character to be replaced, and use the images of different character to yield multiple samples. Here the character image for a given type is selected by measuring the style similarity between the base image and character image.

Following this way, we finally produce 60,000 synthetic license plate images together with their groundtruth segmentation for model training. Fig.5 provides some samples. It can be seen that the synthetic images are very similar to the real ones, which are hard to be distinguished even by human beings. In particular, we also utilize t-SNE, a strong tool for feature visualization, to reveal the data distribution. The results show that the synthesis dataset presents the similar distribution to our Chinese dataset.



**Fig. 5. Samples of synthetic license plate images.** All of them are produced using a random strategy. Here the first character to denote the province is masked for each license plate due to the privacy. Best viewed in color

**Character sequence inference** After obtaining the semantic map of a license plate, we can infer an initial character sequence, which actually represents the label sequence of semantic area along the horizontal direction. Formally, we firstly transform the semantic map  $L$  into  $C$  class-specific binary map  $L'_c$  with a switch function, where  $c \in (1, C)$  and  $C$  is the number of character classes.

$$L'_c(x, y) = \begin{cases} 1, & L(x, y) = c \\ 0, & L(x, y) \neq c \end{cases}$$

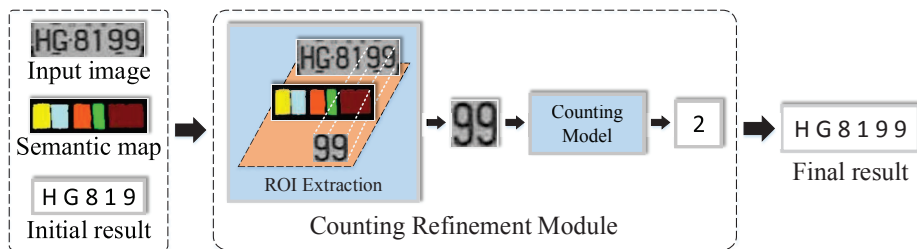
For a binary map  $L'_c$ , we remove the noisy regions of small areas. Then we conduct Connected Component Analysis (CCA) on  $L'_c$  to generate the character areas  $\{A_{ci}\}$ , where  $A_{ci}$  denotes the  $i^{th}$  area in  $L'_c$ .

The wrong prediction of semantic segmentation may make the inference confused and unreasonable. In this work, we propose the following method to improve the robustness. Specifically, we check the Intersection-over-Union (IOU) between the current character area and all previous areas separately. If the IOU exceeds a threshold  $T_f$  ( $T_f = 0.5$  in our implementation), these two character areas are considered to be overlapped. Such case in theory is unreasonable since the characters in a license plate are always arranged separately. We extract the conflicting areas as ROIs, and then apply a classifier (e.g., Inception-v3 model [30]) to determine the final character class. This situation happens occasional in practice, but the proposed method is proven important to correct the hard samples.

### 3.3 Counting refinement module

The counting refinement is used to produce the final character sequence from the raw license plate image, semantic map, and initial character sequence. The main procedures are illustrated in Fig.6. In our LPR framework, it is difficult for semantic segmentation to accurately distinguish the successive character instances of the same class, since the neighboring characters in a license plate are very close to each other and the background region to separate them is too thin for precise recognition. Consequently, multiple characters of the same class are probably regarded as one character in the initial character sequence. To address this issue, we propose to count the characters for such regions.





**Fig. 6. Illustration of Counting Refinement Module.** Our counting refinement module extracts the character region of interests and then applies a classifier to predict the character count. Best viewed in color

Object counting [4, 20] is one of typical visual tasks which aims to obtain the number of specific objects in a given image ROI. In this work, we crop the image regions containing multiple characters of the same class according to the semantic segmentation results as the inputs of the counting model. Considering the application scenario of LPR, the number of characters in a license plate is always limited and even fixed (e.g., 7). Thus we formulate the character counting into a classification problem instead of the regression problem in previous works [23]. Note that the label in this task denotes the number of characters rather than the character class in traditional image classification. Then, we can directly employ some sophisticated classification networks. In this work, AlexNet [19] is particularly adopted due to its high computational efficiency.

In practice, it is observed that the characters '1' and '7' are often misclassified due to their simple contours. Through analysis, it is seen that resizing the input images, which may contain different number of characters in a large range, makes the similar characters harder to distinguish. To handle this issue, we particularly utilize the fact that the region label has been recognized by semantic segmentation. To be specific, we add one white line on the middle of "1" images and two white lines on the middle of "7" images to enhance their appearance difference. This small trick can effectively increase the counting performance.

## 4 Experiment

In this section, we experimentally evaluate the proposed LPR method. Here three challenging license plate datasets are used, *i.e.*, AOLP [14], Media Lab [3], and our own dataset (Chinese License Plate Dataset). For performance metrics, we adopt the recognition accuracy of whole license plate, which is different from the character classification since the recognition result of a license plate would be considered false as long as one of characters is wrongly predicted. For each public dataset, we provide the performance comparison with state-of-the-art methods.

#### 4.1 Implementation details

All of our models are implemented in Caffe [17], and trained/tested on a Tesla K80 GPU. We use a mini-batch of the size 16 at training time. We adopt the Inception-v3 [30] as the character classification model, and AlexNet [19] as the character counting model. Throughout the experiments, the license plate images are resized to  $50 \times 160$ , whose aspect ratio is close to that of physical license plates. For each dataset, the semantic segmentation model is pre-trained on our synthetic license plate images, and then fine-tuned using specific images provided by the dataset. To prevent overfitting, we adopt the common data augmentation strategies, such as noising, blurring, rotation, and color jittering. The semantic segmentation model and character counting model are trained separately. Detailed training protocols are listed in Table 1.

**Table 1.** Training protocols for semantic segmentation and character counting.

	base_lr	batchsize	lr_policy	gamma	step	max_iter
Semantic Segmentation	$10^{-4}$	20	multistep	0.1	[30k, 50k]	60k
Character Counting	$10^{-2}$	512	multistep	0,1	[3k, 5k]	10k

#### 4.2 AOLP dataset

The application-oriented license plate benchmark database (AOLP) collects 1874 images of Taiwan license plates, which is the largest public license plate dataset to our knowledge. The dataset is divided into three subsets according to the usage scenarios, including access control (AC), law enforcement (LE), and road patrol (RP). Specifically, there are 681 images in AC, 582 images in LE, and 611 images in RP. The images in these subsets present different appearance characteristics (*e.g.*, perspective), and thus it is challenging to accurately recognize every license plate. We compare our proposed method with state-of-the-art methods, including Deep FCN [32], LSTMs [21], and Hsu [15].

In the experiment, we use the same training/test split as in LSTMs [21], *i.e.*, two subsets are used for training and the remaining one is for test. We conduct three experiments with different subset for test, and then calculate the recognition accuracies on three subsets and their average accuracy. Table 2 provides the results, where AC/LE/RP denotes the corresponding subset is used for test.

From the results in Table 2, it can be seen that our proposed LPR method achieves the best performance with the accuracies of more than 99% on all subsets. Besides, it is well demonstrated that our method is robust as the images in different AOLP subsets are highly varying in appearance. In particular, our method outperforms Hsu [15] by a 12.32% accuracy improvement, which is mainly due to the elimination of character detection. Compared with LSTMs [21]

**Table 2.** Recognition accuracies on AOLP Dataset. Our proposed method achieves the accuracies of more than 99% in all subsets

	AC (%)	LE (%)	RP (%)	Avg (%)
Hsu (2013) [15]	88.50	86.60	85.70	86.93
LSTMs (2016) [21]	94.85	94.19	88.38	92.47
DeepFCN (2016) [32]	97.90	97.60	98.20	97.90
Our method	<b>99.41</b>	<b>99.31</b>	<b>99.02</b>	<b>99.25</b>

and Deep FCN [32], which adopt a sliding window profile, our method is still superior with an accuracy gain of 6.78% and 1.35%, which experimentally indicates the advantages of the proposed semantic segmentation.

In the above experiment, the recognition performance is evaluated using the subset split. However, high appearance variance in different subsets certainly will decrease the accuracy. To more fully explore the potential of our method, we conduct another experiment using a homogeneous split. Specifically, we evenly divide each subset into three parts with a random strategy. Then we collect two parts of each subset to form the training set and the remaining images are for test. Consequently, three new test subsets are built for performance evaluation, which contain 625, 625, and 624 images, respectively. Table 3 gives the experimental results for the settings. As expected, our method achieves higher recognition accuracies (up to 99.79% for average, *i.e.*, only 4 plates are failed).

**Table 3.** Recognition accuracies of our method with new settings on AOLP Dataset

	Set1 (%)	Set2 (%)	Set3 (%)	Avg (%)
Our method	<b>99.84</b>	<b>99.68</b>	<b>99.84</b>	<b>99.79</b>

### 4.3 Media Lab dataset

NTUA Media Lab dataset consists of 706 Greek license plate images, which covers normal condition and some difficult conditions, *e.g.*, blocking by shadows and extreme illumination. It is divided into a normal subset with 427 samples and a difficult subset with 279 samples. We take Anagnostopoulos [2] and Siyu Zhu [35] as our baselines as they are the state-of-the-art methods on this dataset.

The dataset does not provide the data split for training and test. For fair comparison, we use the images in the normal subset as the test data to evaluate the performance, just like the baseline methods. In our experiment, specifically, we evenly divide the normal subset into four parts with a random strategy. We then conduct four experiments, in each of which three parts with the difficult subset are used for training and the rest one is for test. Table 4 gives the



**Fig. 7. Example images in Chinese License Plate Dataset.** It can be seen that the images present high varying appearance, including illumination, blurring, color, etc. Here the first character is masked due to the privacy. Best viewed in color

experimental results on Media Lab Dataset. It can be seen that our method outperforms the baselines with an accuracy improvement of more than 10%, which again indicates the superiority of the proposed framework. The average accuracy of 97.89% means that only 9 license plates are failed in total.

**Table 4.** Recognition accuracy on Media Lab Dataset.

	Set1(%)	Set2(%)	Set3(%)	Set4(%)	Avg(%)
Anagnostopoulos [2]	-	-	-	-	86.00
Siyu Zhu [35]	-	-	-	-	87.33
Our method	97.14	98.13	98.10	98.18	<b>97.89</b>

In addition, we perform another statistical analysis on the AOLP and Media Lab datasets to demonstrate the robustness of our algorithm to the predictions of semantic segmentation. As shown in Table 5, the license plates can be recognized correctly even when the semantic segmentation module produces imperfect results in pixels.

**Table 5.** Statistics on the proportion of *Wrongly Predicted Pixels* by the semantic segmentation model in the character regions that are finally recognized correctly (denoted as *WPP*). The first three columns shows the *Max*, *Min*, *Mean WPP* in all the samples, while the last three columns are averaged on the Top- $K$  ( $K=5\%$ ,  $10\%$ ,  $20\%$ ) samples sorted by the *WPP* in a descending order.

	Max	Min	Mean	Top 5%	Top 10%	Top 20%
AOLP	38.04%	0.41%	9.21%	24.10%	21.39%	18.32%
Media Lab	35.98%	0.86%	5.53%	16.69%	13.17%	10.46%

#### 4.4 Chinese License Plate dataset

Finally, we evaluate the proposed method on our own dataset, a Chinese License Plate dataset. The dataset is built by collecting the images from the traffic

surveillance cameras, where the license plate images are obtained by applying the SSD [22] detector to raw images. Compared with the public license plate datasets, our dataset is much larger and contains 5057 license plate images. Particularly, the dataset covers diverse conditions, *e.g.*, high varying illumination, motion blurring, and low resolution. Some example images in our dataset are shown in Fig.7. According to the specification of Chinese license plates, each license plate image in the datasets contains a sequence of 7 characters. In particular, there are totally  $(m + 34)$  character classes, including 10 digits, 24 English letters ('O' and 'I' are excluded), and  $m$  Chinese characters<sup>1</sup>.

In our experiments, we randomly split the dataset into two subsets with 4057 images for training and 1000 images for test. Note that the data used for image synthesis are all from the training set. On this dataset, our proposed method achieves a outstanding recognition accuracy of 99.7%. That is, only three license plates in the test set are failed. In addition, our method achieves real-time performance in the computational efficiency (more than 25 FPS on a Nvidia Titan X GPU).

#### 4.5 Error study















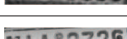

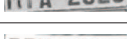

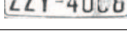

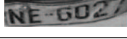





In this section, we conduct an error study to more intuitively demonstrate the performance achieved by our proposed method. In particular, two public datasets, *i.e.*, AOLP and Media Lab, are both used and all failed samples in the above experiments are analyzed. To be specific, the results of 4 failed samples on AOLP and 9 failed samples on Media Lab are provided, as shown in Fig.8. Here we list the raw images, generated semantic maps, ground truth, predicted character sequence, and cause of failure.

From Fig.8, it can be observed that the failure is mainly caused by the poor quality of raw images. For example, the failure that "8" in the first row is recognized as "B" is due to the severely tilt and blurring of license plate, which even cannot be right recognized by a human being. Other failed samples have similar reasons. Overall, the probable causes of failure include image blurring, blocking patches, and severe interference. For these extremely hard samples, our method can still recognize most of characters correctly. Considering the failure cases in error study, we think the performance of our proposed LPR method is very close to the human level.

## 5 Conclusion

In this paper, we proposed a novel license plate recognition framework, which not only achieves outstanding recognition performance but also has high computational efficiency. In particular, our proposed framework introduces the semantic segmentation technique followed by a counting refinement module to accomplish

<sup>1</sup> In the current version of our dataset,  $m = 3$  is used due to the limitation of image acquisition.

Image	Segmentation	Ground Truth	Predicted	Cause of Failure
<b>AOLP Dataset</b>				
		687733	6B7733	blurry condition semantic segmentation mistake
		0750J0	D750J0	'0' is blocked semantic segmentation mistake
		7250EK	7250FK	'E' is incomplete semantic segmentation mistake
		0750J0	D750J0	'0' is blocked character classifier mistake
<b>Media Lab Dataset</b>				
		M1B6822	M1B692	extremely blurry condition
		M010312	H016312	extremely blurry condition
		B183886	B1B3886	blurry condition semantic segmentation mistake
		M1A2326	M1A2328	'6' is blocked semantic segmentation mistake
		ZZY4006	ZZY4066	'0' is blocked semantic segmentation mistake
		NEG027	NE6027	'G' is distorted character classifier mistake
		YBB7799	YBB7795	'9' is blocked semantic segmentation mistake
		YYX4317	YNX4317	blurry condition semantic segmentation mistake
		YXZ5790	YKZ5790	'X' is distorted character classifier mistake

**Fig. 8. Error study on AOLP Dataset and Media Lab Dataset.** Here the processing component causing failure (semantic segmentation or character classifier) is also provided for each sample. Best viewed in color

the recognition of a whole license plate. The experimental results on benchmark datasets demonstrate that our method consistently outperforms previous state-of-the-art methods with a remarkable accuracy improvement. In particular, the error study indicates that the performance of our proposed method is very close to the human level, which makes our method applicable to real systems even for complex scenes. In the future works, we plan to add more components into the proposed framework to support more types of license plates.

## 6 Acknowledgment

This work is supported partially by the NSFC under Grant 61673362, Youth Innovation Promotion Association CAS, and the Fundamental Research Funds for the Central Universities.

## References

1. Adorni, G., Bergenti, F., Cagnoni, S.: Vehicle license plate recognition by means of cellular automata. In: IV (1998)
2. Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., Loumos, V., Kayafas, E.: A license plate-recognition algorithm for intelligent transportation system applications. *IEEE Transactions on Intelligent Transportation Systems* (2006)
3. Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., Psoroulas, I.D., Loumos, V., Kayafas, E.: License plate recognition from still images and video sequences: A survey. *IEEE Transactions on Intelligent Transportation Systems* (2008)
4. Arteta, C., Lempitsky, V., Noble, J.A., Zisserman, A.: Interactive object counting. In: ECCV (2014)
5. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017)
6. Bulan, O., Kozitsky, V., Ramesh, P., Shreve, M.: Segmentation-and annotation-free license plate recognition with deep localization and failure identification. *IEEE Transactions on Intelligent Transportation Systems* (2017)
7. Cheang, T.K., Chong, Y.S., Tay, Y.H.: Segmentation-free vehicle license plate recognition using convnet-rnn. *arXiv preprint arXiv:1701.06439* (2017)
8. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018)
9. Cherng, S., Fang, C.Y., Chen, C.P., Chen, S.W.: Critical motion detection of nearby moving vehicles in a vision-based driver-assistance system. *IEEE Transactions on Intelligent Transportation Systems* (2009)
10. Davies, P., Emmott, N., Ayland, N.: License plate recognition technology for toll violation enforcement. In: *Image Analysis for Transport Applications* (1990)
11. Du, S., Ibrahim, M., Shehata, M., Badawy, W.: Automatic license plate recognition (alpr): A state-of-the-art review. *IEEE Transactions on Circuits and Systems for Video Technology* (2013)
12. Duan, T.D., Du, T.H., Phuoc, T.V., Hoang, N.V.: Building an automatic vehicle license plate recognition system. In: RIVF (2005)
13. Hegt, H.A., De La Haye, R.J., Khan, N.A.: A high performance license plate recognition system (1998)
14. Hsu, G.S., Alexandra, P., Chen, J.C., Yeh, F., Chen, M.H.: License plate recognition for categorized applications. In: ICVES (2011)
15. Hsu, G.S., Chen, J.C., Chung, Y.Z.: Application-oriented license plate recognition. *IEEE Transactions on Vehicular Technology* (2013)
16. Huang, Y.S., Weng, Y.S., Zhou, M.: Critical scenarios and their identification in parallel railroad level crossing traffic control systems. *IEEE Transactions on Intelligent Transportation Systems* (2010)
17. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACM MM (2014)
18. Kim, K.K., Kim, K., Kim, J., Kim, H.J.: Learning-based approach for license plate recognition. In: *Neural Networks for Signal Processing* (2000)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)

20. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: NIPS (2010)
21. Li, H., Shen, C.: Reading car license plates using deep convolutional neural networks and lstms. arXiv preprint arXiv:1601.05610 (2016)
22. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016)
23. Liu, X., Wang, Z., Feng, J., Xi, H.: Highway vehicle counting in compressed domain. In: CVPR (2016)
24. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
25. Masood, S.Z., Shu, G., Dehghan, A., Ortiz, E.G.: License plate detection and recognition using deeply learned convolutional neural networks. arXiv preprint arXiv:1703.07330 (2017)
26. Naito, T., Tsukada, T., Yamada, K., Kozuka, K., Yamamoto, S.: Robust license-plate recognition method for passing vehicles under outside environment. IEEE Transactions on Vehicular Technology (2000)
27. Nijhuis, J., Ter Brugge, M., Helmholt, K., Pluim, J., Spaanenburg, L., Venema, R., Westenberg, M.: Car license plate recognition with neural networks and fuzzy logic (1995)
28. Omitaomu, O.A., Ganguly, A.R., Patton, B.W., Protopopescu, V.A.: Anomaly detection in radiation sensor data with application to transportation security. IEEE Transactions on Intelligent Transportation Systems (2009)
29. Parizi, S.N., Targhi, A.T., Aghazadeh, O., Eklundh, J.O.: Reading street signs using a generic structured object detection and signature recognition approach. In: VISAPP (2009)
30. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
31. Wang, F., Man, L., Wang, B., Xiao, Y., Pan, W., Lu, X.: Fuzzy-based algorithm for color recognition of license plates. Pattern Recognition Letters (2008)
32. Wu, Y., Li, J.: License plate recognition using deep fcn. In: ICCSSP (2016)
33. Yamaguchi, K., Nagaya, Y., Ueda, K., Nemoto, H., Nakagawa, M.: A method for identifying specific vehicles using template matching (1999)
34. Yu, M., Kim, Y.D.: An approach to korean license plate recognition based on vertical edge matching (2000)
35. Zhu, S., Dianat, S., Mestha, L.K.: End-to-end system of license plate localization and recognition. JEI (2015)