

Received March 15, 2020, accepted April 27, 2020, date of publication May 14, 2020, date of current version May 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994287

# Research on License Plate Recognition Algorithms Based on Deep Learning in Complex Environment

WANG WEIHONG<sup>1</sup> AND TU JIAOYANG<sup>2</sup>

<sup>1</sup>Department of Computing, Zhejiang University of Technology, Hangzhou 310000, China

<sup>2</sup>School of Computing, Zhejiang University of Technology, Hangzhou 310000, China

Corresponding author: Tu Jiaoyang (2111712008@zjut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61340058, and in part by the Natural Science Foundation of Zhejiang Province under Grant LZ14F020001.

**ABSTRACT** License plate recognition systems are widely used in modern smart cities, such as toll payment systems, parking fee payment systems and residential access control. Such electronic systems are not only convenient for people's daily life, but also provide safe and efficient services for managers. License plate recognition algorithm is a mature but imperfect technology. The traditional location recognition algorithm is easily affected by light, shadow, background complexity or other factors, resulting in the failure to meet the application of real scenes. With the development of deep learning, the license plate recognition algorithm can extract deeper features, thus greatly improving the detection and recognition accuracy. Therefore, this paper discusses the application of deep learning in license plate recognition, and the main work is as follows: 1) Introduce the most advanced algorithms from the three main technical difficulties: license plate skew, image noise and license plate blur; 2) According to the process, the deep learning algorithms are classified into direct detection algorithms and indirect detection algorithms, and the advantages and disadvantages of the current license plate detection algorithms and character recognition algorithms are analyzed; 3) The differences in data sets, workstation, accuracy and time of different license plate recognition systems are compared; 4) Compare and illustrate the existing public license plate datasets according to the number of pictures, resolution and environmental complexity, and make a prospect for the future research direction of license plate recognition.

**INDEX TERMS** Image processing, image analysis, image classification, license plate recognition, deep learning.

## I. INTRODUCTION

With the rapid development of economy around the world, some cities in different countries may face with traffic congestion, frequent accidents, traffic environment deteriorating or other urban traffic problems. Along with the increase in car use, the following problems in the process of car also gradually emerge, such as: car theft case [1], traffic accident, road congestion, serious environmental pollution, and so on. In order to solve these problems, each country is actively studying how to manage and monitor vehicles more effectively. If only rely on human resources such as traffic police, it will bring many problems such as high cost and low efficiency. Therefore, if intelligent traffic equipment can

be introduced, it will undoubtedly bring great convenience and advantages. In this context, the research on Intelligent Transport System (ITS) [2] was born. The concept of "smart city" arises as the times require, "intelligent video surveillance" and "intelligent transportation" are gradually put on the research agenda to realize intelligent management of cities. License plate recognition system has been widely used in vehicle access management, expressway toll management, intelligent parking, electronic police and other aspects, which plays an important role in the supervision of vehicles, and can realize the supervision of urban traffic to prevent traffic jams, has important significance in real life. At present, there are many license plate recognition systems, but in the complex environment (such as lighting conditions, distorted license plate, dirt license plate, etc.), their license plate recognition rates will be greatly reduced. So how to improve the license

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Cusano.

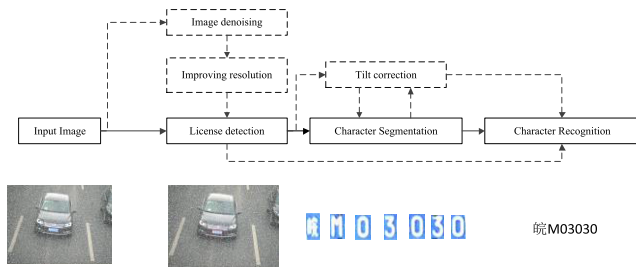


FIGURE 1. Process of license plate recognition system.

plate detection accuracy and recognition rate in the complex environment [3] has great research significance.

The license plate recognition system has two main tasks, one is to locate the license plate, and the other is to identify the license plate characters. The process is shown in figure 1, which is generally divided into three steps, namely license plate location, character segmentation and character recognition. Recently, some people have proposed a two-stage algorithm combining with sequence recognition, therefore the recognition process can only be divided into license plate location and character recognition, leaving out character segmentation. Character segmentation [4] is often used in traditional text recognition algorithms, which use prior knowledge such as fixed character spacing, connected component analysis and project-based methods to implement segmentation. However, due to the handcraft features often can not be used to accurately segment, so the segmentation-free algorithm like sequence labeling can effectively avoid the character segmentation error affecting the recognition accuracy. For noisy scenes, some researches denoise the image and improve the resolution before the license plate detection. For some scenes with skewed shooting angles, some researches have proposed to use the tilt correction algorithms to correct the license plates or the segmented characters to improve the recognition rate.

With the application of license plate recognition technology has become more and more widely, people enjoy the convenience of license plate recognition technology, but can not avoid many difficulties in the process of license plate recognition. The research content of license plate recognition has changed from conventional pictures to complex environments. The main challenges are divided into three aspects:

1) License plate deflection: Most of the datasets used in the early license plate recognition studies are similar to Caltech Car [5] and English LP [6]. These datasets are relatively simple, for example, most images were captured with a handheld camera and there is only one vehicle (generally well-centered) in each image. However, recent datasets, such as Chinese LP [7] and UFPR-ALPR [8], include multiple cars in each image, with the unfixed shooting angle and distance, and the license plate has different degrees of deflection.

2) Noisy plate images: In the real scene, rain, snow and other weather will inevitably occur. In these conditions, some license plates will be blurred and blocked by rain or snow,

while some license plates have uneven lighting. The background is no longer a simple parking lot without other vehicles, but a place with complicated lines just like street intersections with people coming and going.

3) Fuzzy license plates: Freeway monitoring is one of the major applications of license plate recognition system, which often capture the image containing fast moving vehicles. And these vehicles images are usually small object images, for example, in  $1597 \times 1197$  size vehicles images, the boxes size of vehicle are usually around  $533 \times 522$ , nearly 14% of the whole image and the boxes of the license plates are usually around  $123 \times 36$ , nearly 0.2% of the whole image. Additionally, with the high-speed movement of vehicles the license plate characters may become fuzzy. Therefore, there are many researches about how to effectively improve the image resolution of the designated area.

These complicated factors largely restrict the accuracy and reliability of license plate detection and recognition. In order to improve the speed and accuracy of license plate image detection and recognition in complex environments, relevant algorithms need to be studied and improved for one or more factors, which is of great significance for the improvement and optimization of license plate detection and recognition technology. The review of the existing license plate recognition [9], [10] only includes the research on the traditional algorithm, which detects and recognizes according to the intuitive characteristics easily affected by the environment, far from meeting the application of real scenes. Therefore, this paper is devoted to exploring the research of license plate recognition based on deep learning. Through deeper feature extraction, the algorithm can meet the requirements of robustness and real-time under certain detection accuracy.

License plate detection based on deep learning can be divided into direct location and indirect location according to the process. Direct positioning can transform the problem into a general target detection model, commonly used as SSD [11], YOLO [12] and their improved versions [13], [14], and RCNN [15] and their improved versions [16], [17], and so on. Only need to be adjusted into the corresponding datasets and change the number of filters in the last layer to match the number of classes. Aiming at the special context of license plate recognition, some researchers put forward indirect detection algorithm to improve the detection accuracy. Based on the fact that each license plate contains only one license plate, we know that if the picture contains only one car, there must be a license plate. If more than one vehicle is included, there is a certain distance between the two complete license plates. Therefore, the license plate can be located indirectly by detecting the easily detected objects related to the license plate through prior knowledge.

Some license plate recognition systems design character segmentation before character recognition. Segmentation methods can be divided into Connection Component analysis [18], Projection analysis [19], Character prior knowledge [20], Character contour [21] and its combination [22]. As it is known to all, it is difficult to get the correct characters

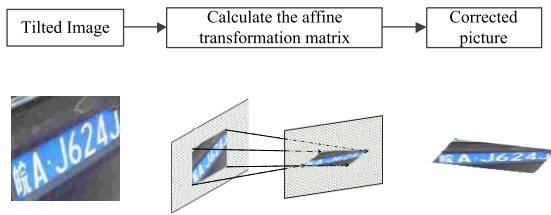


FIGURE 2. Flow chart of license plate deviation correction.

classification of incorrect segmentation results even with high-precision classifier. Therefore, some researchers focus on finding reliable segmentation methods, while others propose segmentation-free method which converts the problem into sequence labeling problem [23] to avoid unstable character segmentation.

The purpose of this paper is to provide researchers with a systematic survey of the existing license plate recognition algorithms, and categorize these algorithms according to the process, compare advantages and disadvantages of detection algorithms and recognition algorithms respectively. In addition, compare different recognition systems in term of the models and datasets and workstation and recognition performance and processing speed as well as whether use the three image processing methods. Other than that, compare different public available license plate datasets in terms of amount and resolutions and give descriptions to each dataset in terms of shooting angle and backgrounds and so on. And finally, explore the future research directions. The remainder of this paper is organized as follows. In the second section introduces the three typical treatments for the three challenges in real situations. The third section introduces the existing license plate location algorithms and the fourth section introduces the existing character recognition algorithms. Finally, the fifth section summarizes this paper and discusses areas for future research.

## II. DIFFICULTIES IN LICENSE PLATE RECOGNITION TECHNOLOGY

### A. LICENSE PLATE DEFLECTION

License plate tilt consists of vertical tilt, horizontal tilt and both. These tilts will undoubtedly result in character distortion and adversely affect character recognition. Therefore, if the pose and part deformation of the object can be disentangled from the texture and shape, it will facilitate the subsequent prediction, for example, local max-pooling layers in CNNs. The process of license plate correction can be regarded as the process of affine transformation, which needs to find out a mapping from tilted image to corrected image. The distorted image is transformed into the corrected image through affine matrix. How to find the affine transformation matrix is the key problem of the skew correction, and the modeling process is shown in figure 2.

Jaderberg *et al.* [24] proposed a method to automatic compute affine transformation matrix by Spatial Transformer

Networks (STN). Once the input images was obtained, which could learn invariance to translation, scale, rotation and more generic warping. Unlike pooling layers, where the receptive fields are fixed and local, the spatial transformer module is a dynamic mechanism that can actively spatially transform an image (or a feature map) by producing an appropriate transformation for each input sample. The spatial transformer module combines the localization network and sampling mechanism. The input of the localization network is feature map  $U$ , and regresses the transformation parameters  $\theta$ ,  $\theta = f_{loc}(U)$ . The localization network function can take any form, for example, fully connected network or a convolutional network, but the last layer must be a regression layer to produce the transformation parameters  $\theta$ . Each output pixel can be computed by sampling kernel centered at a particular location in the input feature map, forming an output feature map  $V$ . In this affine case, the point-wise transformation is

$$\begin{pmatrix} x_i^s \\ y_i^s \\ 1 \end{pmatrix} = T_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

where  $(x_i^t, y_i^t)$  and  $(x_i^s, y_i^s)$  both normalize to  $[-1, 1]$ , are the coordinates in the target and source respectively.  $A_\theta$  is the affine transformation matrix, equal to the output of the location network. After the transformation, the source coordinates in the input feature map are applied to get the value at a particular pixel in the output  $V$ . This can be written as

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y) \quad \forall i \in [1 \dots H'W'] \quad \forall c \in [1 \dots C]$$

where  $\Phi_x$  and  $\Phi_y$  are the parameters of a generic sampling kernel  $k()$ , which defines the image interpolation (e.g. bilinear).  $U_{nm}^c$  represents the value at location  $(n, m)$  in channel  $c$  of the input,  $V_i^c$  is the output value for pixel  $i$  at location  $(x_i^t, y_i^t)$  in channel  $C$ .

This kind of spatial transformer can be incorporated into other convolutional neural networks, which effectively improved the representation of deep network and improve the recognition accuracy of convolutional neural network. Therefore this transformer is a method that was trained with tilted images and normal images to automatically find a mapping between two kinds of images, and usually incorporated by many license plate recognition algorithms. Multiple spatial transformers can also be used simultaneously to identify multiple objects in a single image. This space transformers can not only affine the whole license plate, but also affine several characters in the license plate.

### B. IMAGE WITH NOISE

The sources of noise in the image may come from various ways such as image acquisition, transmission and compression, and the types of noise can also be divided into pepper and salt noise, gaussian noise and so on. Usually for the license plate recognition, in the real scene, the license plate

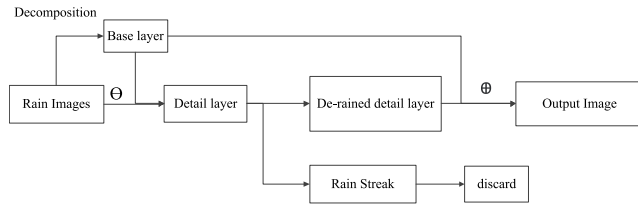


FIGURE 3. Flow chart of image denoising.

image will distort suffered by rain line, snow line and other noises, and some license plates might be defaced. There is no good solution for image denoising. In practice, it is more about to achieve a tradeoff between effect and computational complexity. Some denoising algorithms directly use the values of adjacent pixels to calculate the average value, such as the bilinear filtering and median filtering, while others regard the noisy image as the superposition of noise and clear image, decompose the image into the detail layer and the base layer, and then further separate the noise streaks from the detail layer by the network. Finally combine the base layer and the de-noising detail layer to get the output image. The modeling process is shown in figure 3, where  $\ominus$  denoting subtraction operation and  $\oplus$  express add operation.

Kang *et al.* [25] removed rain streaks from a single image by image decomposition, who regarded rain streaks removal as a problem of image decomposition by morphological component analysis. First, the image is divided into high and low frequency parts by bilateral filtering. Then the high frequency part is decomposed into a rain component and a nonrain component by performing dictionary learning and sparse coding. Finally got the image preserving most original details.

Li *et al.* [26] proposed an effective method that uses simple patch-based priors to solve the problem of rain streaks removal from a single image. Based on the assumption that the rain-free background layer and the rain streaks layer are independent of each other, the two different GMMs are used to construct the constraints of the background layer and the rain streaks layer respectively. This method is more effective than dictionary learning and low rank constraint method.

Fu *et al.* [27] was used the deep network architecture called DerainNet to remove the rain streaks from single image. First the input rainy image was decomposed into the basic layer and the detail layer. And then the detail layer was fed into DerainNet to get De-rained detail layer. The basic layer and the De-rained detail layer were augmented with the image enhancement to improve the visual result, the output of network was the linear superimposition of the enhanced base layer and the enhanced detail layer.

Yang *et al.* [28] proposed a method used the deep joint rain detection and removal from a single image. The previous work simply built the model as a superposition of clean images and rain streaks. By observing the natural images, they proposed a new component representing the rain streaks accumulation of various shape and directions which visually similar to mist or fog. Therefore, two variables of global

atmospheric light and atmospheric transmission were introduced to rebuild the rain model. This rain model can effectively solve the limitations of over-smoothing the regions of the most existing algorithm, and meet various rain conditions in the real scene. In order to obtain more contexts while preserving rich local details other than limited in a local image patch, they used a contextualized network. Then the deep convolutional network based on patch priori [12] was developed to detect and remove the rain streaks, and the network was extended to a recurrent model in consideration of the accumulation of rain and heavy rain.

These denoising algorithms can achieve better visual results, but usually time consuming. In addition to the above denoising algorithms, we compare the performance and time of other algorithms. Since it is difficult to obtain the real rain image and corresponding clean images, the rain images synthesized from clean images are adopted in the experiment. In table 1, there are several evaluation metrics used in these literatures, which SSIM refers to the structural similarity index, and VIF refers to the fidelity of visual information, and PSNR refers to the peak signal to noise ratio. A higher SSIM indicates that the effect of denoising algorithm is closer to the real label value in terms of image structure. And the range of VIF and SSIM are both [0, 1]. The larger the value is, the closer to the clean image.

License plate recognition systems not only require accuracy, but also take processing time into account. In table 1 above, the algorithms [25] and [26] are not suitable for recognition system, because both of them are difficult to meet the real-time requirements. In contrast, the algorithms [29] and [31] can achieve high signal-to-noise ratio and structural similarity with the processing time is around 0.2 seconds, therefore are suitable for combining with the license plate recognition system to improve the recognition accuracy.

### C. FUZZY LICENSE

In the image to be tested, there is only a small proportion of vehicles but large of background, containing pedestrians, houses and sky, especially when captured from a large stand-off distance. As a part of the vehicle, license plate occupies a smaller proportion in the whole image, so license plate recognition can be transformed into a small target recognition problem. However, detecting small objects is notoriously challenging due to their low resolution and noisy representation. Though direct up-sampling via interpolation could be viewed as a possible solution for low resolution recognition, it will be disadvantageous to the subsequent character recognition, or it will be time consuming to detect the small target by learning its representation on multiple scales. Therefore, it is necessary to find a method to improve the resolution of small target license plate effectively.

Li *et al.* [32] was used Perceptual GAN to internally lift representations of small objects to “super-resolved” ones, achieving similar characteristics as large objects and thus more discriminative for detection. Different from the general GAN, the discriminator network has two branches, i.e.

TABLE 1. Performance and time of different denoising algorithms.

Ref	Year	experiment platform	Rational	Metric		Image size	Time (s)
[25]	2012	Intel Core i5-460M, 4GB RAM	bilateral filter, dictionary learning, sparse coding	VIF	$0.55 \pm 0.05$	256*256	94.36
[26]	2016	--	Gaussian mixture models	SSIM	$0.91 \pm 0.05$	480*640	93
[27]	2017	Intel Core i5 CPU 4460 , 8GB RAM , NVIDIA Geforce GTX 750	bilateral filter, deep residual network	SSIM	$0.89 \pm 0.06$	250*250	1.3
						500*500	2.8
						750*750	5.4
[29]	2017	GPU, Caffe	Deep residual network	SSIM	$0.90 \pm 0.05$	250*250	0.2
						500*500	0.3
						750*750	0.5
[28]	2017	GPU	multi-task recurrent CNN model	PSNR	$36.11 \pm 0.09$	80*80	0.11
				SSIM	$0.97 \pm 0.01$	500*500	1.46
[30]	2018	--	generative adversarial network	PSNR	31.57	--	--
				SSIM	0.9023		
[31]	2018	GTX 1080 GPU	Deep Decomposition Composition Network	PSNR	$35.47 \pm 3.09$	250*250	0.03
				SSIM	$0.93 \pm 0.03$	500*500	0.12

adversarial branch and perception branch. The adversarial branch is used to differentiate between the generated super-resolved representation and the original one for the large object. And the perceptual branch is used to justify the detection accuracy benefiting from the generated representation. Finally, the generator network was designed as a deep residual learning network to augment the representations of small objects to super-resolved ones. The loss function of discriminator network is composed of the weighted sum of two parts. The first part of the adversarial branch used the -log loss function. In the second part, the perceptual branch adopts the smooth L1 loss function. The parameters of the generator network were obtained by optimizing the discriminator network loss function. For the difficulty in training the GANs, Tolstikhin *et al.* [33] studied the use of cascade generation model to solve the problem of model absence in training. Radford *et al.* [34] and Salimans *et al.* [35] aim to make the counter generation network more stable and easier to fine-tune.

Singh *et al.* [36] proposed a dual directed capsule network called DirectCapsNet to recognize very low resolution image, which could achieve the recognition accuracy over 95% when 16\*16 images are matched with 80\*80 images. This model combines capsule and convolutional layers to learn the effective low-resolution recognition model. Three the loss functions were incorporated to enhance the edge features, push the low resolution image features of a particular class towards a representative high resolution feature of that class, and force the capsules of low resolution and high resolution images of the same class to be similar. High resolution and corresponding low resolution images were used during training. At the time of testing, the class capsule with the highest length was chosen as the class of the given input.

Similar to the tilted correction technique, the above-mentioned algorithms for improving the resolution is also

to find the mapping relationship between the small object to the large object. Relative to the simple linear transformation increase, the algorithms based on the generated network could add richer details to the pictures. However, due to the difficulty of training generation network and real-time requirements, it may hinder the applications of improving resolution in actual scenarios.

#### D. DISCUSSION

In this section, we discuss some unavoidable problems of license plate recognition in real situations. For weather conditions such as rain, snow and fog, and use times include day and night, as well as license plate images taken with a handheld camera at the roadside that can result in motion blur, wobbling and deflection. The factors that affect the accuracy of license plate recognition include illumination, shielding, the shooting angle, the shooting distance, camera resolution, background complexity and noise and so on. According to these factors, license plate preprocessing is divided into three aspects, that is, tilt correction, image denoising and improving resolution, so as to obtain the ideal image after recovery. Table 2 highlights some studies on these three aspects in relevant license plate recognition literature from 2005 to 2018. Among them, the study of image denoising is to remove the noisy such as rain and snow that block the license plate in the image, and the study of improving the resolution is to restore the blurred image caused by the camera or moving object to the high resolution discriminative image.

The tilted correction network realizes image correction mainly by obtaining affine transformation matrix, which can be inserted into other networks. However, Image denoising is always computational complex. According to the experimental platform and time consumption in table 1, the time required for [25], [26] is more than 90 seconds, which

**TABLE 2. Literature in different research directions from 2005 to 2018.**

Research directions	Refer
Tilt correction	[37][38][39][40][41][42][43][44]
Image denoising	[45][46]
Improving resolution	[47][48][49][50][51][52]

obviously cannot meet the real-time requirements for license plate recognition scenes. Some denoising algorithms, such as [30], do not even make temporal comparison. Reference [28] took advantages of cyclic multi-task convolutional network, which can reach the SSIM score of 0.97, higher than other algorithms, but the required time is almost 5 times that of [29]. Due to the special requirement of real-time in license plate recognition scene, a preprocessing algorithm is needed to improve the image quality as much as possible under the condition of real-time.

According to the literature statistics in table 2, the algorithms to improve license plate recognition results by image denoising are significantly less than those to improve tilt correction and resolution, which also takes into account the tradeoff between accuracy and time. Image denoising algorithms attempt to separate the noise layer from the image, while improving the resolution algorithms look for the functional mapping between the fuzzy image and the clean image. For example, [47] used the blind deconvolution algorithm to reversely find the sharp image of the fuzzy image, with the datasets shot on the road of a speed limit of 90km/h, which could reduce the character error rate from 23% to 9% and increase it by 2.5 times. The processing time needed about 0.5 seconds, which showed a certain gap in real-time. Generative adversarial network can also improve the accuracy of small target recognition, but due to the complexity of the network, it is difficult to train, so how to combine generative adversarial network for end-to-end identification needs to be further studied.

### III. LICENSE PLATE DETECTION

The license plate location stage is to extract a portion area containing the potential license plate from the input car image. Without a license plate location, direct recognition would struggle to distinguish license plate Numbers from other text blocks such as traffic signs and phone Numbers on storefronts, and license plates might occupy only a small portion of the image so that may be ignored. The accuracy of license plate location will affect the subsequent recognition performance. In this stage, the input is the image containing the car, and the output is four values representing the license plate location. The coordinates of the upper left corner and the height and width, namely  $(x, y, w, h)$ , are commonly used to represent. Traditional algorithm based on the manual extraction of features to achieve the location, such as the license plate color features, texture and other single or combined features. However, the prior knowledge is limited, which will lead to the information of the whole picture cannot be fully utilized. Deep learning can extract features based on the pixel

information of input images, and deeper network can extract more detailed features. Moreover, features that are difficult to be extracted manually, such as multi-scale information and fine-grained information, can be obtained through different feature extraction models. The whole network structure usually consists of the feature extraction layer and the parameter regression layer. The regression layer is generally the full connection layer, and the number of neurons is adjusted to the number of prediction parameters, which are 4 in usual.

Traditional license plate location algorithms can be divided into five categories based on the intuitive features: text-based detection, color-based detection, character-based detection, and connected-component detection. These intuitive features are easily affected by environmental, while deep learning can extract deeper features by pixel information. This paper divides the localization algorithms into direct location and indirect location according to the process. Direct location refers to the regression network to directly predict the coordinates of the license plate, as well as the length and width, while indirect location refers indirectly obtain the information of the license plate through other indicators that are easy to locate the vehicle, for example, detect the car or the rear light of the car firstly then calculate the plate coordinates.

#### A. DIRECT LOCATION

Direct location is to directly predict the location and height and width information of license plate by feeding the input picture. For training, the license plate coordinates, height and width should be measured from the picture, and the loss function such as Euclidean distance is utilized to calculate the parameter gradient. Compared with indirect detection, direct detection can save computational cost to some extent, but it is not as accurate as indirect detection.

Kurpiel *et al.* [53] partitioned the input image to 120 pixels wide by 180 pixels high, that way, the sub-regions form an overlapping grid. Then send each sub-region to CNN of nine layers to get a confidence score, and the output value ranges  $[0,1]$ . The further the license plate center moves outside, the lower the score decreases from 1 to 0. Finally, the location of the license plate is estimated by combining the output values of all image sub-regions so that the license plate center is closer to the left or right sub-regions with the highest score. Its detection accuracy can reach 0.87, recall rate is 0.83 and processing time is 0.23 seconds.

Li *et al.* [54] proposed a unified deep neural network for end-to-end training to simultaneously locate and identify license plates. It consists of several convolutional layers, a region proposal network for license plate proposal generation, proposal integrating and pooling layer, multi-layer perceptions for plate detection and bounding box regression, and RNNs with CTC for plate recognition. The feature extraction network is modified according to the vgg-16 network to keep the convolutional layer, reduce the number of pooling layers and abandon the full connection layer. The modified RPN was utilized to output a set of potential bounding boxes, and 6 scales with an aspect ratio of 5 was designed to

generate 6 anchors at each position of the input feature maps. Inspired by inception-RPN [55], two 256-d convolutional filters were applied simultaneously across each sliding position. The extracted features were concatenated along the channel axis to obtain 512-dimensional feature vectors, which were then fed into two separate fully convolutional layers, one for plate/non-plate classification, and the other for box regression. The detection model could realize the accuracy of 98.15% and only needs 300ms to process each image.

Xiang *et al.* [56] presented an efficient lightweight full convolutional network for license plate detection from complex scenes, which downscales input images for substantially accelerating proceeding and reducing the computational cost. In order to further improve the prediction accuracy, dense connections and dilated convolutions are adopted for combining multi-level and multi-scale vision features, and the fusion loss structure is appended during training. A fusion loss structure is appended during training to further improve prediction accuracy. The network consists of two parallel branches. The foreground branch samples the image down to 1/8 of the original image, and the main part of the background branch is built with dense blocks [57], each of which contains a series of connected convolutional layers. A 3\*3 convolution with stride 2 is adopted instead of pooling in each block to subsample feature maps, so as to reduce the calculation cost without losing accuracy. On the Caltech datasets, the detection accuracy could reach 93.47%, and the processing time per image was 28.33ms.

## B. INDIRECT LOCATION

Under the complex environment, sometimes it is difficult to directly locate the license plate, especially when the license plate target is too small or partial shielding. Since the license plate is part of the body of a car, human experience can determine the approximate location of the license plate even if it cannot be located immediately at first glance. Therefore, some researchers take advantage of the prior knowledge between the license plate and the car body, such as the position relationship between the rear lights and the license plate [58], to transform the problem into a target that is easier to detect.

Li *et al.* [59] used a cascaded framework to read the license plate, which first detected the character region and then extracted the license plate frame. To begin with, a 4-layer 37 class CNN classifier is employed in a sliding window fashion across the entire image to detect the presence of the text and generate a text saliency map, the candidate bounding boxes are generated independently at each scale by using the run length smoothing algorithm (RLSA) [60] and connected component analysis (CCA). Then, the generated boxes are filtered by geometric constraints and refined by the edge feature of license plate [61]. Finally, another plate/non-plate CNN classifier was used to verify the remaining bounding box. This detection model could reach the precision about 97% and the recall more than 95%.

Dong *et al.* [39] used a cascade structure composed of a fast region proposal network and a R-CNN network to extract license plate. First, a light-weight RPN network [17] took the down-sampled image as input to generate the license plate candidates. Then the sampler extracts the region of interest (ROIs) from the original high-resolution image. And the extracted patches are fed into the R-CNN network to classify the candidate plate and regress four corners of the license plate. This license plate detector is 1.5 times faster and 57 times smaller than faster R-CNN, and achieved the accuracy more than 96%.

Silva *et al.* [62] used the YOLOv2 network without any change or refinement to detect vehicles, just regard the network as a black box, merging the two classes of cars and buses on the PASCAL-VOC dataset, and ignored other classes. Insights from YOLO, SSD, and STN, the WPOD-NET was proposed to detect license plates in a variety of different distortions, and regresses coefficients of an affine transformation that unwarps the distorted license plate into a rectangular shape resembling a frontal view.

Xie *et al.* [63] proposed a CNN-based method called MD-YOLO inspired by the YOLO framework to realize multi-directional car license plate detection. Similar to YOLO, each input image was divided into regular 7\*7 grid cells, and the cell in which the car license plate center is located is used to detect the license plate, and to predict 2 bounding boxes and a confidence score for each cell. Versus YOLO the difference is MD-YOLO introduces angle information and guides the model to regress and determine the angle of rotation of a given car license plate image. The angle deviation penalty factor (ADPF) was proposed to approximate the intersection ratio between predicted value and tag value. And in order to identify negative rotation angle values, leaky and identity functions were chosen as activation functions, rather than ReLU function. Considering that the license plate is usually very small, a prepositive CNN attention model called ALMD-YOLO was employed prior to the implementation of MD-YOLO. This detection model could achieve more than 99% accuracy with 5ms processing time on GPU GTX980.

Laroca *et al.* [64] implemented the system conceived by evaluating and optimizing different Yolo models with various modifications, aiming at achieving the best speed/accuracy tradeoff at each stage. For license plate detection stage, considering that the license plate might only occupy very small portions of the image, and other textual blocks like traffic signs might confuse with license plates, the same detection process as [62] was adopted, in which the vehicles were first detected, and then detect their respective license plates in vehicle patches. This system could achieve average precision of 98.37% and average recall of 99.92% on 8 different datasets.

## C. DISCUSSION

In this section, the existing license plate detection algorithms are described and divided into direct location and indirect location according to its recognition process. By analyzing

the above algorithms, the accuracy of the direct positioning algorithm is lower than that of the indirect positioning algorithm for most time. For example, the accuracy of [53] is 87% and 93.47% of [56] are both lower than that of the indirect positioning algorithm [39], [59], [63], [64]. In table 3, we summarize above existing detection algorithms and analysis the advantages and disadvantages of each algorithm.

The network models adopted by the above algorithm are listed in table 3. Due to its fast execution (around 70FPS), and good precision and recall compromise (76.8% mAP over the PASCAL-VOC dataset), most literatures use YOLO and other versions to be adapted for license plate detection. The above indirect detection algorithms are all composed of two different networks, which can tolerate a certain degree of light variation, distortion and blur.

#### IV. LICENSE PLATE RECOGNITION

Character recognition is the conversion of license plate images into character sequences. Different from text information extraction, license plate recognition will not be too variance in font and font size, but the use of characters in different countries are very different. Take the Chinese license plate as an example, the first character is a Chinese character, representing the province information. The second character is a capital letter, representing the city information. Each character of the later five or six could be a letter or a number. Especially, the new clean energy license plates have 8 characters, other license plates have 7 characters. The rear license plates of buses and trucks may be in the different layout, Chinese characters and capital letters are on the first line, and the rest combinations of letters and Numbers are on the second line. The goal of character recognition is to output all characters accurately, including the correct classification of each character, and to prevent the occurrence of missing and redundant characters.

The license plate extracted in the previous stage is used as input and the character sequence is output. In the traditional license plate recognition procedure, character segmentation has a great influence on the precision of license plate recognition. A license plate can be misidentified if it is not properly segmented, even if there is a strong recognizer that can handle various zooming, different fonts, and various rotations. Therefore, some researchers consider on how to segment effectively. And some, in order to avoid difficult character segmentation, combined with the recent RCNN, propose a series of sequence labeling methods. In this paper, according to whether character recognition is segmented or not, the recognition algorithm is divided into segmentation-based and segmentation-free recognition algorithm.

##### A. BASED ON THE SEGMENTATION

In terms of segmentation, traditional algorithms can be roughly divided into five categories: connected component analysis, projection analysis, prior knowledge of characters, character contours and their combinations. With the development of deep learning, object detection model is also used

to extract characters. After the license plate is divided into a single character, the recognition algorithm is delivered.

Liu *et al.* [65] implemented the combination of connected component analysis and project analysis for the segmentation. Two simple and recurrent CNNs were designed for character recognition, namely SCNN and RCNN, one for Chinese characters recognition and the other was used to recognize numbers and letters. For overexposed license plates, which cannot be directly treated with binarization process, so a grayscale conversion algorithm was presented in advance, whose basic idea is to suppress the grayscale of color points and enhance the contrast ratio of images. Finally, breadth-first search algorithm is used to obtain the connected components, and the connected components were used to determine whether there were missing or redundant characters. Based on 2189 images, the segmentation rate and recognition rate were 96.58% and 98.09%, respectively.

Dong *et al.* [39] proposed an innovative structure consisted of a parallel spatial transformation networks and a shared-weight recognizers. The corrected license plate was fed into seven parallel unsupervised STNs [24], and each STN implicitly performs character segmentation. Finally, seven recognizers recognize each segment, the first recognizer separately trained for the Chinese character, and the rest six recognizers share weights. For the weight sharing recognizer, the score is obtained by multiplying the prior probability and the likelihood estimated by the recognition sub-network, where the prior probabilities of Arabic numerals digits of the second character were 0. The final recognition accuracy of the model is 89.05%.

Khare *et al.* [66] proposed a new concept called partial character reconstruction, which was introduced to segment characters of license plates to enhance the performance of license plate recognition system. Due to the symmetry properties of character stroke width were sensitive to blur, torching and complex backgrounds, the Canny edge image of the input image is used to find the same symmetry properties with weak conditions in the gray domain to reconstruct the full character shape, so as to improve the recognition rate of characters. After the Laplace operation of the edge image, the width corresponding to the highest peak is selected as the extracted stroke width distance. Based on the discovery that principal component analysis (PCA) and major axis (MA) can estimate the directions of character components without the complete character shape, a method of character segmentation based on angle information was proposed. If the Angle given by PCA and MA is almost the same and both almost 90 degrees, the component is considered as a full character. If the difference between the two axes is more than 26 degrees and the value given is almost 0, the component is considered to be an under-segmentation. Otherwise, the component was considered as a case of over-segmentation. An iterative-shrinking or iterative-expansion algorithm was used for characters that were under-segmentation or over-segmentation. This model achieves a segmentation rate of 82.6% and recognition precision of 87.3%.



**TABLE 3. Feature extraction networks adopted by different positioning algorithms.**

Method	Refer	Year	Rational	Pros	Cons
Direct location	[54]	2017	VGG-16 + Inception RPN	Be able to be end to end train.	Computational complex, when region proposal it centered as each pixel to generate different size anchor boxes.
	[53]	2017	CNN models	Be able to detect under variance light and motion blur with fast execution speed, and easy to be trained.	Lower precision.
	[56]	2019	Parallel network	Lower processing time.	Processing unimportant pixels and vulnerable to distortion plates.
Indirect location	[39]	2017	RPN + R-CNN	Robust to plate rotation, motion blur and variance light.	Computationally complex when region proposal.
	[64]	2018	ALMD-YOLO + MD-YOLO	Be able to detect rotation plate, and detect multiple plates simultaneously.	Each of two models requires separate training, and struggle to detect very small plates.
	[63]	2018	YOLOv2 + WPOD-NET	Be able to detect distorted plates shooting from different angles.	Computationally complex and hardly to be trained end to end.
	[59]	2018	CNN-I + CNN-II	Be able to detect plates of different font size in different countries, and reach high precision.	Feature extraction lost color information, and take time in sliding window fashion.
	[65]	2019	Yolov2 + Fast-Yolov2	Real time detection with high precision.	A large number of data need to manually label

### B. BASED ON THE SEGMENTATION FREE

The algorithms based on the segmentation-free transform the vehicle license plate recognition problem into character sequence labeling. Unlike the segmentation based algorithms, the segmentation-free algorithms utilize the global information of the input image. Hidden Markov Models (HMM) and its hybrid method HMM-RNN are the earliest framework for realizing sequence labeling, but their pre-segmentation and post-processing operations seriously limit their practicability. Therefore, Alex Graves *et al.* proposed a connectionist temporal classification (CTC) model [67] to solve the problem of sequence labeling without segmentation.

Li *et al.* [54] employed bidirectional RNNs (BRNN) with CTC loss to label the sequential feature. Two additional convolutional layers with ReLUs reform the extracted plate region features to a sequence with the size as  $512 \times 19$ . BRNNs consist of two separated RNN layers with 512 units were used to process forward and backward propagation respectively. The two hidden states were concatenated together and fed into a linear transformation with 37 outputs to identify 10 digits and 26 English letters and a special non-character class. The probabilities were recorded at each time step, and after BRNNs encoding, the extracted plate region feature sequence is transformed into a sequence of probability estimation with the same length as feature sequence. To prevent the gradient vanishing or exploding during traditional RNN training, Long-Short Term Memory (LSTM) was used. Finally, CTC layer was adopted for sequence decoding. On the AOLP dataset, the average recognition rate was 91.83% with the execution speed of 400ms.

Zhuang *et al.* [68] transferred the YOLO-VOC network for license plate segmentation and character recognition. Due

to the characteristics of Brazilian license plate, there are 7 characters in total, the first 3 characters are letters and the last 4 characters are numbers, so all detected characters are filtered by two heuristic rules. Through experiments, appropriate input and output scales are selected to balance the speed and accuracy of system recognition. The model can correctly segment 99% of characters, with a recognition rate of 93%, and the execution time on the GPU is only 2.2ms.

Li *et al.* [59], based on the previous literature of [54], trained the recurrent neural networks (RNNs) with LSTM to recognize the sequential features extracted from the whole license plate via CNNs. Each detected license plate was converted to a gray-scale image and resized to  $24 \times 94$  pixels. A sub-window of  $24 \times 24$  pixels with a step size of 1 was used to partition the padded image in a sliding window manner. Each partitioned image patch was fed into 36-class CNN classifier to extract sequence features. The fourth convolutional layer and the first fully connected layer were concatenated together into one feature vector with length 5096. Then PCA was used to reduce the feature dimension to 256 dimensions, followed by the feature normalization. Finally, CTC was designed to decode the predicted probability sequence into output labels directly with average recognition rate of around 92.47%.

Wang *et al.* [40] used BRNN and CTC to recognize the license plates, which the recognition rate was 96.62% with processing time of 17.53ms. Firstly, the spatial transformer network (STN) was employed to adjust the inclined and deformed license plates, and the license plate with the uniform orientation was fed into the improved convolutional neural network (CNN) to extract the sequence features of the rectified license plates. These features of different

convolutional layers were integrated as an input to BRNN, and finally the sequence labels were realized by CTC.

Silva *et al.* [69] proposed a novel license plate recognition system consisted of semantic segmentation and character counting towards achieving human-level performance. Pre-processing of simple projection on the input image was employed to make the images suitable for semantic segmentation. Modified DeeplabV2 ResNet101 model was adopted for semantic segmentation to produce the semantic map of the input license plate image with the same size, whose value represents the character class of corresponding pixel, and initial character sequence. and feeds the cropped image into the improved DeeplabV2 resnet-101 model to predict character classes element by element. Adjacent characters belonging to the same class may be hard to be separated, so counting refinement was proposed to extract the hard to be separated area and send it to the classifier AlexNet to predict the count of characters in such region. Then the final character sequence could be generated, with processing speed is 38FPS and recognition precision over 99% on AOLP dataset.

Table 4 analyzes the main network that each algorithm depends on for the above segmentation and segmentation-free based recognition algorithms, and evaluates the advantages and disadvantages of each algorithm.

### C. DISCUSSION

In this section, the existing character recognition algorithms based on deep learning are described, and the recognition algorithms are divided into segmentation-based and segmentation-free algorithms, and the advantages and disadvantages of the above character recognition algorithms are analyzed by the table 4. Table 5 compares the amount and resolution of the existing public license plate datasets and describes other situations such as occlusion and blur. Among them, CD-HARD is the challenging images picked from the Cars Dataset. In addition, common object detection datasets such as PASAL-VOC, ImageNet and COCO also contain some vehicle classes, but since these datasets are not dedicated license plate datasets, they are not compared in the table 5. Due to different resolution or complex degree images will have an effect on the accuracy of license plate recognition algorithms, therefore, the test accuracy should be evaluated objectively according to the test datasets. For example, datasets like SSIG, extensive use of front view of the vehicle, and centered image without a large angle deflection, that can achieve high accuracy in most of the algorithms. However, AOLP and other datasets contain a large number of images of license plates such as uneven illumination and skew, which will increase the difficulty of recognition.

Table 6 compares the different classical license plate recognition algorithms based on deep learning in terms of the models, datasets, precision, processing time and whether used image processing to raise accuracy. For example, the classic commercial system Sighthounds proposed by Masood *et al.* [76] has a smaller limitation than the OpenALPR system, and can be identified without entering the country of license

plate. Due to the limited public datasets of license plates, network training requires a large number of pictures and corresponding labeling data. Therefore, some algorithms use private datasets downloaded from the Internet or taken in real scenes for training and testing. Considering the effect of experimental equipment on processing speed, the workstation used in each algorithm is explained. In literature [64], if a system can process at least 30 frames per second (FPS), i.e. the average processing time per image is about 33ms, then the system meets the real-time requirement, because commercial cameras usually record video at this frame rate [8], [77], [78]. Some algorithms use public datasets for testing and training, while others also use private datasets for testing and training to increase the amount of samples. In table 6, BW represents the license plate with white characters on a blue background, while YB represents the license plate with black characters on a yellow background. In the list of stage, D represents detection, S represents segmentation, and R represents recognition. Datasets, precision and processing time in each stage are listed. The number of images used is listed in parentheses at the end of the dataset, and finally, processing refers to whether used license plate correction, denoising, and resolution enhancement to improve recognition accuracy.

As can be seen from table 6, most of the recent algorithms based on deep learning adopt the recognition algorithms without segmentation, which directly recognizes the license plates instead of the character patches of the intermediate results, so as to avoid the intermediate errors. It is proposed that only LPR-Net [88] is the unified end-to-end model, and other algorithms can realize end-to-end recognition, but it is not a unified model. Among them, six algorithms used special processing, most of which carried out license plate correction, and one of which denoised the image. Compared with Chinese characters and Arabic, which are more complex in number and letter structure, they are more sensitive to the influence of various factors in real scenes, and these kind of distorted and fuzzy characters are also harder to be recognized by the naked eyes, so correction and recovery are more needed.

### V. SUMMARY AND FUTURE DIRECTIONS

The process of license plate recognition is usually divided into three steps, that is, license plate detection, character segmentation and character recognition. In the license plate recognition system, in order to deal with the complex situations such as uneven illumination, unfixed shooting angle, different weather conditions and motion blur in the real scenes, different processing algorithms are proposed in order to repair the image to the one that could be detected and recognized by the system. As input images acquired in these different complex scenes, the license plate processing algorithms are divided into three categories according to the image restoration technology, that is, license plate correction, denoising and high resolution representation, and some typical processing of these algorithms are also described. In the license plate detection stage, some researchers explore

**TABLE 4. Comparison of advantages and disadvantages of different character recognition algorithms based on deep learning.**

method	Refer	Year	Rational	Pros	Cons
Segmentation based	[66]	2018	SCNN+RCNN	Be able to recognize under various illumination	Be sensitive to distorted and occlusion conditions
	[39]	2017	RANSAC+STN+CNN	Weight sharing used in the method could make full use of training dataset	Unable to suitably segment new energy and two-row layout license plates
	[68]	2019	PCA+MA	Be able to reconstruction characters of blur and occlusion to enhance the recognition precision	Over-segmentation when meeting Chinese characters with left-right structure
Segmentation-free based	[69]	2017	FAST-YOLO	Fast	Low precision
	[54]	2017	BRNN+CTC	Be able to end-to-end train, and robust to various illumination	Distorted and occlusion affect the recognition precision
	[59]	2018	RNN+CTC	Make full use of contextual information, and get high precision rate.	Unable to be realize in real-time
	[40]	2018	STN+CNN+BRNN+CTC	Be able to train end-to-end, and simultaneously recognize numbers and Chinese characters	Limited to the assumption of the layout, unable to recognize plates in two rows
	[70]	2018	Modified-Deeplab-V2-ResNet-101-model+CCA+AlexNet	Recognition by pixel-wise so robust to individual miss prediction	Fixed count of characters fail to recognize other layout plates like new energy plates

**TABLE 5. Comparison of public available license plate datasets.**

Dataset	Year	Amount	Resolution	Country	Description
Caltech Cars[5]	1999	126	896*592	American	Simple, one vehicle centered at each image, vehicle patch occupy a large proportion, easy to identify
EnglishLP[6]	2003	509	640*480	European	Simple, one vehicle centered at each image, easy to identify
USCD-Still[71]	2005	291	640*480	American	Simple, only one vehicle in each image, easy to identify
ChineseLP[7]	2012	411	Variance	Chinese	Multiple cars in each image, unfixed shooting angles, more difficult to identify
AOLP[72]	2013	2049	Variance	Chinese	Shielded, skewed and uneven illumination plates, RP is the most difficult subset of the three to identify
CD-HARD[63]	2016	102	Variance	Variance	Multiple country plates with multiple layout, clear images, but large tilt shooting angle, difficult to identify
OpenALPR-EU[73]	2016	108	Variance	European	Certain deflection angle, uniform illumination, easy to identify
SSIG-SegPlate[74]	2016	2000	1920*1080	Brazil	Clear images, distant shooting, easy to identify
PKU Dataset[75]	2017	3977	Variance	Chinese	Uneven illumination, skewed plates, subset G1-G4 only one car in one image but multi cars in G5, difficult to identify
UFPR-ALPR[8]	2018	4500	1920*1080	Brazil	Far distance shooting, unfixed shooting angle, uneven illumination, difficult to identify
CCPD[76]	2018	200K	720*1160	Chinese	Multiple scenario subsets, one vehicle in each image, blurred and skewed plate, difficult to identify
GAP-LP[8]	2019	9175	Variance	Tunisia	Multiple layout, variance shooting angle, uneven illumination, difficult to identify

indirect detection algorithms to improve the recall rate because of the influence of light intensity, blurring, occlusion and distortion. And with the study of sequence labeling algorithms, the segmentation-free license plate recognition process is proposed, which only includes license plate loca-

tion and character recognition. In this paper, the advantages and disadvantages of license plate detection algorithms and recognition algorithms are analyzed. In addition, the datasets of public license plate are sorted out to compare the amount, image resolution and illustrate the complexity of the image

TABLE 6. Comparison of license plate recognition algorithms based on deep learning.

Ref	Year	Stage	Model	Datasets		Workstation	Precision (%)	Time (ms)	Process
				Training	Testing				
[77]	2017	D	Deep detection CNN	License plate images(>20K)	American LP[80][5] (348) European LP [6][73](608)	--	99.36	--	No
		S	Binary deep network	Cropped license plate(>25K)			94		
		R	Deep	License plate					
			recognition CNN	characters(>100K) Negatives(>100K)					
[39]	2017	D	RPN+R-CNN	PC Train_Det(12.7K)	PC Eval_Det(5.9K)	NVIDIA K40 GPU	96.68	90.9	Yes
		S	STN	PC Train_Rec(4.6K)	PC Eval_Rec(1.1K)		89.05		
		R	CNN						
[66]	2018	D	Color-based+Sobel	--	PC BW(2189) PC YB(200)	Intel Core i5-3210M, 8GB RAM	98.95	237	No
		S	CCA+P A		Successfully located BW(2166)		96.58	25	
		R	SCNN+RCNN		Chinese characters(1046) Alphanumeric characters(6276)		Successfully segmented (2092)	98.09	
[59]	2018	D	CNNs+CCA	Character images[60][81] (138k) Non-character images cropped from ICDAR[82][83][84][85] (900k) License plate and the synthesize[86][87] (453k)	Caltech Cars AOLP	NVIDIA Tesla K40c GPU, 6GB RAM	97.28	3000	No
		R	BRNN+CTC	Character images[60][81] (138k)	AOLP		92.47		
[88]	2018	D	RPN	ImageNet Dataset	Standard dataset(10837) Real scene dataset(471)	--	88.83	37.5	No
		R	Fast R-CNN	Extracted objects			97.5		
[63]	2018	D	YOLOv2+WPOD-NET	Cars Datasets[89](105) SSIG(40) LE subset of AOLP(51)	OpenALPR (BR and EU)(222) SSIG AOLP CD-HARD(10)	NVIDIA Titan X GPU, 12GB RAM	89.33	200	Yes
		R	OCR-NET[70]	Synthetic and augmented plate					
[40]	2018	D	EasyPR	--	PC(800)	Intel core i7-6850k, NVIDIA GeForce GTX 1080Ti, 11GB RAM	96.62	17.53	Yes
		R	STN+CNN+BRNN+CTC	Synthetic image(100k) PC(1520) Augmented image(7600)					
[90]	2018	D	LPR-Net	Collected images(2000)	Night-set(410) Askew-set and fuzzy-set(410)	GeForce GTX 1080 GPU	99.8	200	Yes
		R					95.8		
[65]	2019	D	YOLOv2+Fast-YOLOv2	Caltech Cars(78) EnglishLP(407) UCSD-Stills(220) AOLP(1366) SSIG-SegPlate(1196) ChineseLP(238) UFPR-ALPR(2700)	Caltech Cars(46) EnglishLP(102) UCSD-Stills(60) ChineseLP(159) AOLP(683) OpenALPR-EU (108) SSIG-SegPlate(804) UFPR-ALPR(1800)	NVIDIA Titan Xp GPU	99.45	11.63	No
		R	CR Net	Extracted LP patches	96.8		1.99		
[41]	2019	D	Plate detector network	Positive synthetic images(22.5K) Negative synthetic images(22.5K)	AOLP PKU dataset	GeForce GTX 1080 GPU	98.9	20.1	Yes
		R	Character detector network		AOLP		96.4	5.4	
[91]	2019	D	YOLOv2+BRNN+CTC	GAP-LP(7573) Radar LP Dataset(4448)	GAP-LP(1602) Radar LP Dataset(2000) AOLP(2049)	NVIDIA GTX 1080 Ti GPU, 64GB RAM	97.67	36.7	No
		R					95.88	7.6	
[92]	2020	D	Mask-	True LP(256K)	AOLP(2049)	NVIDIA	98.7	--	No

TABLE 6. (Continued.) Comparison of license plate recognition algorithms based on deep learning.

		RCNN		Non-LP(140k)		PKU Dataset(3977) Caltech(126) Extension Tunisian(1350)	GeForce GPU, 8GB RAM	97.93		
		S	R	Characters(90K) Non-characters(50K)						
[93]	2020	D	SegNet [94]	Collected plates(450)	Collected plates(150)	NVIDIA Quadro M4000 GPU, 64GB RAM	94.43	--	Yes	
		S	OSTU				91.67			
		R	CNNs							

from the aspects of lighting conditions and shooting distance and so on. In addition, compare some classical license plate recognition systems based on deep learning are compared in term of the models, datasets, workstation, accuracy, processing time, and whether used correction, denoising and high resolution repressing to improve accuracy.

License plate recognition technology is a mature but imperfect technology. Although the license plate recognition system appeared in the 1990s, the algorithms at that time only considered in simple and single scenes and got low recognition efficiency. There are also a number of recent algorithms that claim to achieve higher accuracy and lower computational complexity, but only for some specific datasets. In order to measure the algorithm whether satisfies the application in real life, the test set containing multiple scenes should be tested and combined with multiple evaluation indicators. Table 6 lists the different algorithms based on deep learning and their datasets, and provides the source of the datasets. According to the data in table 6, the most advanced recognition algorithm at present is the improved model based on YOLO proposed by [64], which can achieve the recognition accuracy of 96.8% for multiple scenes in multiple countries, while only 13.62ms is required to process each image. However, the literature of [20] only improved the model to efficiently detect the license plate, but did not correct or denoising to enhance the performance of precision, so there is still room for improvement.

In general, the existing license plate recognition algorithms can be improved according to the following aspects:

1) Existing target detection algorithms can achieve better detection performance to big target object but to small target is not ideal. However, the small target plate more sensitive to blur and occlusion and other interference, so the future algorithms can combine with image deblurring and license plate correction or improving the resolution of the small target, in order to improve the rate of license plate detection and subsequent character recognition precision.

2) The diversification of the evaluation system and the test set that contain various scenes may take contributions to objectively evaluate the system. As the public available datasets are limited, as shown in table 5, there are only 126 pictures of Caltech Cars, only 291 pictures of USCD-Still, and the most is 4500 of UFPR-ALPR. However, too few datasets are not conducive to deep learning training and test evaluation. Therefore, on the one hand, we should make full

use of multiple datasets; on the other hand, we should look for large amounts datasets, such as CCPD [65], which contains nearly 200,000 pictures and can be divided into subsets of different conditions, including fuzzy set, rotating set, tilted set and different weather state set, and so on.

3) Most existing system are trained by two or three models, which means that the corresponding datasets should be collected and labeled before training, and the matrix parameters of corresponding models should be downloaded before testing, which undoubtedly causes certain labor cost and computer storage cost for system deployment. Therefore, there is an urgent need to find a system that can be end-to-end trained and tested with a unified model.

Based on the current research progress, this paper present a comprehensive survey on existing license plate systems based on deep learning algorithms, and categorize the algorithms at each stage by the process. The advantages and disadvantages of detection and recognition algorithms are compared respectively. And the different license plate recognition systems based on deep learning are compared in term of models, datasets, precision and processing time. In addition, some public available license plate datasets are sorted out to compare the amount of each dataset and image resolution, and explain the situation in terms of shooting angle, illumination conditions, and other background complexity. Finally, future forecasts for license plate systems are given at the end, which should be concentrated on solving the three aspects of complex scenes, namely, license plate correction, denoising and high resolution representation, as well as the diversified evaluation system and the construction of a unified model to be end-to-end trained and tested.

REFERENCES

- [1] C. Lum, J. Hibdon, B. Cave, C. S. Koper, and L. Merola, "License plate reader (LPR) police patrols in crime hot spots: An experimental evaluation in two adjacent jurisdictions," *J. Experim. Criminology*, vol. 7, no. 4, pp. 321-345, Dec. 2011.
- [2] 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.
- [3] Z. Sanyuan, Z. Mingli, and Y. Xiuzi, "Car plate character extraction under complicated environment," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2004, pp. 4722-4726.
- [4] M.-S. Pan, J.-B. Yan, and Z.-H. Xiao, "Vehicle license plate character segmentation," *Int. J. Autom. Comput.*, vol. 5, no. 4, pp. 425-432, Oct. 2008.
- [5] M. Weber, P. Perona. (1999). *Caltech Cars Dataset*. [Online]. Available: [http://www.vision.caltech.edu/Image\\_Datasets/cars\\_markus/cars\\_markus.tar](http://www.vision.caltech.edu/Image_Datasets/cars_markus/cars_markus.tar)

- [6] V. Srebric. (2003). *EnglishLP Database*. [Online]. Available: [http://www.zemris.fer.hr/projects/LicensePlates/english/baza\\_slika.zip](http://www.zemris.fer.hr/projects/LicensePlates/english/baza_slika.zip)
- [7] 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.
- [8] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Goncalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the YOLO detector," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–10.
- [9] C. Patel, D. Shah, and A. Patel, "Automatic number plate recognition system (ANPR): A survey," *Int. J. Comput. Appl.*, vol. 69, no. 9, pp. 21–33, 2013.
- [10] 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.
- [11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
- [12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [13] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.
- [14] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [16] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [17] 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.
- [18] B.-F. Wu, S.-P. Lin, and C.-C. Chiu, "Extracting characters from real vehicle licence plates out-of-doors," *IET Comput. Vis.*, vol. 1, no. 1, pp. 2–10, Mar. 2007.
- [19] W.-Y. Ho and C.-M. Pun, "A macao license plate recognition system based on edge and projection analysis," in *Proc. 8th IEEE Int. Conf. Ind. Informat.*, Jul. 2010, pp. 67–72.
- [20] J.-M. Guo and Y.-F. Liu, "License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques," *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1417–1424, May 2008.
- [21] A. Capar and M. Gokmen, "Concurrent segmentation and recognition with shape-driven fast marching methods," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2006, pp. 155–158.
- [22] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, and T. Shiose, "A novel adaptive morphological approach for degraded character image segmentation," *Pattern Recognit.*, vol. 38, no. 11, pp. 1961–1975, Nov. 2005.
- [23] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014, *arXiv:1409.3215*. [Online]. Available: <https://arxiv.org/abs/1409.3215>
- [24] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," 2015, *arXiv:1506.02025*. [Online]. Available: <http://arxiv.org/abs/1506.02025>
- [25] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012.
- [26] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2736–2744.
- [27] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2944–2956, Jun. 2017.
- [28] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1685–1694.
- [29] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1715–1723.
- [30] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, "Attentive generative adversarial network for raindrop removal from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2482–2491.
- [31] S. Li, W. Ren, J. Zhang, J. Yu, and X. Guo, "Fast single image rain removal via a deep decomposition-composition network," 2018, *arXiv:1804.02688*. [Online]. Available: <https://arxiv.org/abs/1804.02688>
- [32] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1951–1959.
- [33] I. O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf, "AdaGAN: Boosting generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5424–5433.
- [34] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016, *arXiv:1511.06434v2*. [Online]. Available: <https://arxiv.org/abs/1511.06434>
- [35] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training GANs," *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
- [36] M. Singh, S. Nagpal, R. Singh, and M. Vatsa, "Dual directed capsule network for very low resolution image recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 340–349.
- [37] L. Liu, S. Zhang, Y. Zhang, and X. Ye, "Slant correction of vehicle license plate image," in *Proc. Int. Conf. Image Anal. Process.*, 2005, pp. 237–244.
- [38] B. Yanpin and Y. Bai, "Application of radon transformation to license plate slant correction," *J. Test Meas. Technol.*, vol. 23, no. 5, pp. 452–456, 2009.
- [39] M. Dong, D. He, C. Luo, D. Liu, and W. Zeng, "A CNN-based approach for automatic license plate recognition in the wild," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–12.
- [40] J. Wang, H. Huang, X. Qian, J. Cao, and Y. Dai, "Sequence recognition of chinese license plates," *Neurocomputing*, vol. 317, pp. 149–158, Nov. 2018.
- [41] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, "Robust license plate recognition using neural networks trained on synthetic images," *Pattern Recognit.*, vol. 93, pp. 134–146, Sep. 2019.
- [42] K. Deb, A. Vavilin, J.-W. Kim, and K.-H. Jo, "Vehicle license plate tilt correction based on the straight line fitting method and minimizing variance of coordinates of projection points," *Int. J. Control, Autom. Syst.*, vol. 8, no. 5, pp. 975–984, Oct. 2010.
- [43] P. Li, M. Nguyen, and W. Q. Yan, "Rotation correction for license plate recognition," in *Proc. 4th Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2018, pp. 400–404.
- [44] M. Y. Arafat, A. S. M. Khairuddin, and R. Paramesran, "A vehicular license plate recognition framework for skewed images," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 11, pp. 5522–5540, 2018.
- [45] S. Yoshimori, Y. Mitsukura, M. Fukumi, N. Akamatsu, and R. Khosal, "License plate detection system in rainy days," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom. Intell. Robot. Autom. for New Millennium*, Jul. 2013, pp. 972–976.
- [46] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi, "An algorithm for license plate recognition applied to intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 830–845, Sep. 2011.
- [47] P. Svoboda, M. Hradis, L. Marsik, and P. Zemcik, "CNN for license plate motion deblurring," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3832–3836.
- [48] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 769–777.
- [49] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 257–265.
- [50] J. Fang, Y. Yuan, W. Ji, P. Tang, and Y. Zhao, "Licence plate images deblurring with binarization threshold," in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, Sep. 2015, pp. 1–6.
- [51] Q. Lu, W. Zhou, L. Fang, and H. Li, "Robust blur kernel estimation for license plate images from fast moving vehicles," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2311–2323, May 2016.
- [52] P. S. Prashanth Rao and R. Kumar Muthu, "A new de-blurring technique for license plate images with robust length estimation," 2018, *arXiv:1802.06214*. [Online]. Available: <http://arxiv.org/abs/1802.06214>
- [53] F. Delmar Kurpiel, R. Minetto, and B. T. Nassu, "Convolutional neural networks for license plate detection in images," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3395–3399.
- [54] H. Li, P. Wang, and C. Shen, "Towards End-to-End car license plates detection and recognition with deep neural networks," 2017, *arXiv:1709.08828*. [Online]. Available: <http://arxiv.org/abs/1709.08828>
- [55] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, "DeepText: A unified framework for text proposal generation and text detection in natural images," 2016, *arXiv:1605.07314*. [Online]. Available: <https://arxiv.org/abs/1605.07314>

- [56] H. Xiang, Y. Zhao, Y. Yuan, G. Zhang, and X. Hu, "Lightweight fully convolutional network for license plate detection," *Optik*, vol. 178, pp. 1185–1194, Feb. 2019.
- [57] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.
- [58] M. R. Asif, Q. Chun, S. Hussain, M. S. Fareed, and S. Khan, "Multinational vehicle license plate detection in complex backgrounds," *J. Vis. Commun. Image Represent.*, vol. 46, pp. 176–186, Jul. 2017.
- [59] H. Li, P. Wang, M. You, and C. Shen, "Reading car license plates using deep neural networks," *Image Vis. Comput.*, vol. 72, pp. 14–23, Apr. 2018.
- [60] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 512–528.
- [61] L. Zheng, X. He, B. Samali, and L. T. Yang, "An algorithm for accuracy enhancement of license plate recognition," *J. Comput. Syst. Sci.*, vol. 79, no. 2, pp. 245–255, Mar. 2013.
- [62] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 593–609.
- [63] L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, "A new CNN-based method for multi-directional car license plate detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 507–517, Feb. 2018.
- [64] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector," 2019, *arXiv:1909.01754*. [Online]. Available: <https://arxiv.org/abs/1909.01754>
- [65] Y. Liu, H. Huang, J. Cao, and T. Huang, "Convolutional neural networks-based intelligent recognition of chinese license plates," *Soft Comput.*, vol. 22, no. 7, pp. 2403–2419, Apr. 2018.
- [66] V. Khare, P. Shivakumara, C. S. Chan, T. Lu, L. K. Meng, H. H. Woon, and M. Blumenstein, "A novel character segmentation-reconstruction approach for license plate recognition," *Expert Syst. Appl.*, vol. 131, pp. 219–239, Oct. 2019.
- [67] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn. ICML*, 2006, pp. 369–376.
- [68] J. Zhuang, S. Hou, Z. Wang, and Z.-J. Zha, "Towards human-level license plate recognition," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 314–329.
- [69] S. Montazzoli and C. Jung, "Real-time brazilian license plate detection and recognition using deep convolutional neural networks," in *Proc. 30th SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2017, pp. 55–62.
- [70] L. Dlagnekov and S. Belongie. (2005). *UCSD Dataset*. [Online]. Available: [http://vision.ucsd.edu/belongie-grp/research/carRec/car\\_data.html](http://vision.ucsd.edu/belongie-grp/research/carRec/car_data.html)
- [71] 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.
- [72] OpenALPR Inc. (2016). *OpenALPR-EU Dataset*. [Online]. Available: <https://github.com/openalpr/benchmarks/tree/master/endtoend/eu>
- [73] G. R. Gonçalves, S. P. G. da Silva, D. Menotti, and W. R. Schwartz, "Benchmark for license plate character segmentation," *J. Electron. Imag.*, vol. 25, no. 5, Oct. 2016, Art. no. 053034.
- [74] 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.
- [75] Z. Xu, W. Yang, A. Meng, N. Lu, H. Huang, C. Ying, and L. Huang, "Towards end-to-end license plate detection and recognition: A large Dataset and baseline," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 261–277.
- [76] S. Z. Masood, G. Shu, A. Dehghan, and E. G. Ortiz, "License plate detection and recognition using deeply learned convolutional neural networks," 2017, *arXiv:1703.07330*. [Online]. Available: <https://arxiv.org/abs/1703.07330>
- [77] G. R. Gonçalves, M. A. Diniz, R. Laroca, D. Menotti, and W. R. Schwartz, "Real-time automatic license plate recognition through deep multi-task networks," in *Proc. 31st SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2018, pp. 110–117.
- [78] H. Li, P. Wang, and C. Shen, "Towards End-to-End text spotting with convolutional recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5248–5256.
- [79] M. Hill. *Openalpr-U.S.*. Accessed: Jul. 2014. [Online]. Available: <https://github.com/openalpr/benchmarks/tree/master/endtoend/us>
- [80] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1457–1464.
- [81] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, 2003, pp. 682–687.
- [82] S. M. Lucas, "ICDAR 2005 text locating competition results," in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2005, pp. 80–85.
- [83] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: Reading text in scene images," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 1491–1496.
- [84] A. Criminisi. (2004). *Microsoft Research Cambridge Object Recognition Image Database*. [Online]. Available: <http://research.microsoft.com/en-us/downloads/b94de342-60dc-45d0-830b-9f6eff91b301/default.aspx>
- [85] C.-N.-E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 377–391, Sep. 2008.
- [86] Z. Yang, F.-L. Du, Y. Xia, C.-H. Zheng, and J. Zhang, "Automatic license plate recognition based on faster R-CNN algorithm," in *Proc. Int. Conf. Intell. Comput.*, 2018, pp. 319–326.
- [87] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 554–561.
- [88] D. Wang, Y. Tian, W. Geng, L. Zhao, and C. Gong, "LPR-Net: Recognizing chinese license plate in complex environments," *Pattern Recognit. Lett.*, vol. 130, pp. 148–156, Feb. 2020.
- [89] Y. Kessentini, M. D. Besbes, S. Ammar, and A. Chabbouh, "A two-stage deep neural network for multi-norm license plate detection and recognition," *Expert Syst. Appl.*, vol. 136, pp. 159–170, Dec. 2019.
- [90] Z. Selmi, M. B. Halima, U. Pal, and M. A. Alimi, "DELP-DAR system for license plate detection and recognition," *Pattern Recognit. Lett.*, vol. 129, pp. 213–223, Jan. 2020.
- [91] N. Omar, A. Sengur, and S. G. S. Al-Ali, "Cascaded deep learning-based efficient approach for license plate detection and recognition," *Expert Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113280.
- [92] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–9.



**WANG WEIHONG** was born in Zhejiang, China, in 1976. He received the master's degree in computer application from Zhejiang University, in June 1999.

He has been Teaching with the Zhejiang University of Technology, since 2000, where he is currently a Professor with the School of Computer Science. He has published more than 40 articles in important journals and conferences at home and abroad, including more than 30 articles in three major indexes. His research interests include image and graphics processing, remote sensing and geographic information systems, and information security.

Prof. Weihong received the Third Prize from the Zhejiang Province Science and Technology Progress Award.



**TU JIAOYANG** was born in Zhejiang, China, in 1994. She is currently pursuing the master's degree in computer science with the Zhejiang University of Technology.

Her research interests include image processing and deep learning.

...