Robust Automatic Recognition of Chinese License Plates in Natural Scenes

MING-XIANG HE and PENG HAO

1College of Computer Science and Engineering, Shandong University of Science and Technology, Tsingtao 266590, China
2National Virtual Simulation Experiment Center of Shandong University of Science and Technology, Shandong University of Science and Technology, Tsingtao 266590, China

Corresponding author: Peng Hao (haopeng2018@sdust.edu.cn)

This work was supported in part by the Industry-University Cooperation Education Program of the Ministry of Education under Grant 201901035016.

ABSTRACT Automatic license plate recognition has a wide range of applications in intelligent transportation systems and is of great significance. However, most of the current work on license plate recognition focuses on the images on the front of license plates. License plate recognition in natural scenes and arbitrary perspective is still a huge challenge. To solve this problem, this work mainly studies the detection and recognition of inclined Chinese license plates in natural scenes. We propose a robust method that can detect and correct multiple license plates with severe distortion or skewing in one image and input them into the license plate recognition module to obtain the final result. Different from the existing methods of license plate detection and recognition, our method performs affine transformation during license plate detection to rectify the distorted license plate image. It can not only avoid the accumulation of intermediate errors but also improve the accuracy of recognition. As an additional contribution, we put forward a challenging Chinese license plate recognition data set, including images obtained from different scenes under a variety of weather conditions. Through a large number of comparative experiments, we have proved the effectiveness of our proposed method.

INDEX TERMS Convolutional neural networks, deep learning, license plate detection, license plate recognition.

I. INTRODUCTION

With the latest developments in intelligent transportation and deep learning (DL), Automatic License Plate Recognition (ALPR) has become an important frontier field of research [1]. Most of the current traffic applications, including traffic flow monitoring and parking lot access verification, involve license plate recognition (LPR) that is performed by an ALPR system [2]. ALPR is expected to become a key issue in the field of computer vision in future intelligent cities. In future intelligent cities, the camera network deployed at road intersections will recognize vehicles driving in the city in real-time [3]. Generally, ALPR consists of 3 steps: vehicle detection, license plate detection (LPD), and LPR [4], [5], [30]).

The first step of ALPR is vehicle detection. In a real natural environment, the system first captures the vehicle in the scene. This often requires a powerful object detection algorithm [6]–[8]. If the vehicle is not detected in the first step, it will directly affect the subsequent series of work. The second step is to detect the license plate (LP) from the detected vehicles. Edge detection [9] is a commonly used detection method in LPD. Except for edge detection, many methods have been proposed to address this problem [10]–[15]. Finally, we recognize the LP number of the detected license plate. Generally, the LPR methods are divided into two types: one is the character segmentation method [16], and the other is the end-to-end method [17]–[20].

Though technology is improving, most ALPR systems still use a front view of LPs. As these schemes depend on specific constraints, they often result in poor generalization performance of the system and cannot be widely applied.
Whereas, LPs in natural scenes may appear skewed due to the angle of the lens—although people may still recognize it with the naked eye (as shown in Figure 1), which is the main challenge facing us.

Our research contributions can be summarized as three main points:

1) We proposed a new method of LPs detection that can accurately detect severely skewed or distorted LPs, and then performed an affine transformation on the detected LPs to obtain the simulated front view images.

2) We reconstructed a robust three-level LPR network framework. In each level, we can choose different network structures to build LPR network according to our needs.

3) To make our work more convincing, we manually collected a batch of seriously skewed, but still readable Chinese LPs as our test set. The annotations of the data set will be publicly available so that it can be used as a new and challenging benchmark for ALPR research.

The rest part of this paper is structured as follows:

In Chapter 2, we briefly review the related ALPR methods; in Chapter 3, we detailly introduce the method proposed in this paper; in Chapter 4, we introduce our experimental in detail and the final experimental results; and finally, we provide a summary of our research work and a prospect for future work in Chapter 5.

II. RELATED WORK

ALPR is the task of finding and recognizing LPs in natural scenes. It is usually divided into three sub-tasks. They form a system of 3 sequential modules: vehicle detection, license plate detection, and license plate recognition.

The foundation of ALPR is a powerful detection method. Experts and scholars have conducted extensive research in this field. Many different ALPR methods or their related sub-tasks have been proposed in the past [21]–[23]. Compared with vehicles, the object of the license plates is relatively small and hard to detect.

A. LICENSE PLATE DETECTION

The previous researchers usually used manual features and classic machine learning classifiers, [24] with image binarization and grayscale analysis to fix the license plate area. Their mainstream methods can be divided into three categories: edge-based, color-based, and texture-based. Yuan et al. [25] proposed a new linear density filtering method for the connection of the areas with high edge density. Then, the sparse areas of each row and column in the binary edge image are removed. Chang et al. [26] proposed a detection method for LPs in Taiwan on the basis of different foreground and background colors on RGB images. They designed a color edge detector that is sensitive to the edges between black and white, red and white, and green and white. Yu et al. [27] focused on the License plate detecting issues under the conditions of large changes in illumination and background. They proposed a License plate location method on the basis of wavelet transform and EMD analysis, which showed good accuracy in the license plate location. However, these traditional methods mentioned above have high computational complexity. The actual effect will be disturbed by the environment, and the processing time is too long, so they are not applicable to real-time LPD.

With the rise of DL [28], advanced technologies begin to move in the direction of DL. Nowadays, many studies
use deep CNNs for LPD. He et al. [29] proposed a residual learning framework for the solution of a deep network that is hard to train. This innovative method makes the features extracted by the network more sufficient and can obtain higher accuracy. Li et al. [30] proposed a network based on Faster R-CNN. A region suggestion network is used to fix the area of the candidate LP and cut out the feature maps accordingly through the RoI pooling layer. Then, these objects of candidate LPs are input to the last part of the network to work out the probability of whether it is an LP. The success of the YOLO network has inspired many recent studies on License plate detection. Xie et al. [31] put forward a CNN-based MD-YOLO framework for multi-directional LPD. It adopts a rotation angle prediction and intersection-over-union evaluation strategy to realize high-precision real-time LPD. Laroca et al. [32] put forward a robust ALPR system based on the YOLO object detector. For each ALPR stage, CNNs are trained and fine-tuned (for example, changes in cameras, lights, and background) to ensure they are robust under complicated conditions. Whereas, it is not easy for the YOLO network to detect small-sized objects, so we need to further evaluate the scene where vehicles are away from the camera. The recent works [33], [34] also take an image rectification approach and explore spatial transformer networks [35] for scene text distortion correction. Similarly, [36], [37] integrate the rectification and detection into the same network. These recent systems exploit deep convolutional networks for correction and show very promising detection performance.

B. LICENSE PLATE RECOGNITION

How to correctly identify LPs on the basis of correctly detection of them is also an important issue. What related to LPR is Scene Text Recognition (STR), which aims to find and read text/numbers in natural scenes. The existing STR work can be divided into two categories. One is to use a bottom-up approach, which prioritizes the detection and recognition of individual characters. The other is to use a top-down approach to directly identify words or text lines. Most traditional STR systems use a bottom-up approach. Firstly, the detection and recognition of individual characters has a certain manual feature. Then, the recognized characters are connected to words by using certain methods. Such as the use of sliding windows [38], [39], connected components [40], extremal regions [41], Hough voting [42], co-occurrence histograms [43]. But most of them are limited by the ability of manual feature representation. With the development of DL in recent years, various CNN-based methods have been designed for scene character recognition. For example, [44] the using of a fully connected network for character recognition, [45] the using of CNNs for feature extraction, and [46] that of CNNs for character recognition in unconstrained environments. These deep network-based methods need to locate a single character. Due to the complex image background, there is too much contact between adjacent characters. It is often resource-intensive and prone to errors. To solve the above problems, researchers have proposed various top-down methods. We can directly recognize entire words or lines of text without the need to detect and recognize individual characters. Besides, Recurrent Neural Networks (RNNs) have also been extensively studied. It encodes words or text lines into a feature sequence, which can be recognized without character segmentation. For example, [47], [48] converts extracted cross-text sequences to feature sequences using RNNs. [19], [49], [50] use RNNs for visual feature representation and CTC for sequence prediction. In recent years, attention mechanisms have gradually become popular. Improve the recognition ability by detecting more discriminating and informative areas. For example [51], a novel character attention mechanism has been designed for end-to-end STR. Recent studies [52], [53] have used image correction methods in the exploration of spatial transformation networks for scene text distortion correction. It has good robustness and flexibility in estimating and correcting text distortions.

III. PROPOSED METHOD

In this chapter, we will describe in detail the working principle of our proposed method. The overall framework of this method is shown in Figure 2. The proposed method mainly includes three modules:

1) Vehicle Detection
2) License Plate Detection
3) License Plate Recognition

In the following sections, we will detailly introduce each module.

A. VEHICLE DETECTION

Considering that the vehicle is one of the underlying objects in many classical detection algorithms and object detection data sets (such as PASCAL-VOC [54], ImageNet [55], and COCO [56]), we decided not to design or train the detector from the ground up, but to select a known mature algorithm to perform vehicle detection.

When choosing an object detection algorithm, we considered the following issues: First, the algorithm requires a high accuracy rate. Because any missing detection vehicle will directly cause the overall License plate detection to be missed. Second, the algorithm requires a high calculation speed. If the high calculation speed is not reached, it will cause the phenomenon of stuttering and affect real-time detection. Finally, the calculation cost is low, and it can be widely used. The huge model size and expensive computing cost will hinder their deployment in many real-world applications.

After careful consideration, we decided to use EfficientDet [57] as our vehicle detection network, because it performs fast, and the accuracy and calculation efficiency are very good. We did not make any modifications or improvements to EfficientDet but used the network as a black box to merge the output of related classes and ignore other classes that are not related to traffic scenarios.
**B. LICENSE PLATE DETECTION**

Considering the shooting angle in the real environment, we proposed a distortion correction LPD network. The LPD network structure for distortion correction is shown in Figure 3. The network can learn to detect LPs in natural scenes with different degrees of distortion, and correct the distorted LPs into a rectangular shape similar to the front view.

The size of the license plate is smaller than that of the entire scene. To locate the license plate more accurately, we designed a network based on residual blocks to extract features. All the convolution filters are fixed in size at $3 \times 3$. To prevent the license plate feature information from disappearing after pooling, we only used four $2 \times 2$ maximum pooling layers with a step-size of 2, which reduces the input dimension by 16 times. Finally, the detection block has two parallel convolutional layers: (1) one for interring the probability, activated by a softmax function, and (2) another for regressing the affine parameters, without activation.

The process of extracting LPs using a distortion correction network is shown in Figure 4. First, the network generates a feature map with a total of 8 channels for object/non-object probabilities (2-channel) and affine-transformation parameters (6-channel). To extract the characters on the distorted LPs, we firstly considered a hypothetical square whose size is fixed at the LP center $(i, j)$. If the object probability of the
FIGURE 4. Distortion correction process.

unit is greater than the given detection threshold, the regression parameters are used to establish an affine matrix to convert the virtual square into the License plate area. Therefore, we can easily extend LPs to horizontally and vertically aligned objects.

Let \( m_i = [x_i, y_i]^T \), for \( i = 1, 2, 3, 4 \), denoting the four corners of an annotated License plate, clockwise starting from top-left. Also, let \( n_1 = [-0.5, -0.5]^T \), \( n_2 = [0.5, -0.5]^T \), \( n_3 = [0.5, 0.5]^T \), \( n_4 = [-0.5, 0.5]^T \) denote the corresponding vertices of a canonical unit square centered at the origin.

For an input image with height \( H \) and width \( W \) and network stride given by \( N_s = 2^4 \) (4 max-pooling layers), the network output feature map consists of \( P \times Q \times 8 \) volumes, where \( P = H/N_s, Q = W/N_s \). For each point cell in the feature map, there are 8 values to be estimated: the first 2 values \((v_1, v_2)\) are the object/non-object probabilities, and the last 6 values \((v_3 \text{ to } v_8)\) are used to establish the local affine transformation \( T_{ij} \) given by:

\[
T_{ij}(n) = \begin{bmatrix}
\max(v_3, 0) & v_4 \\
v_5 & \max(v_6, 0)
\end{bmatrix} n + \begin{bmatrix} v_7 \\ v_8 \end{bmatrix}
\]

(1)

\( T_{ij}(n) \) uses the learned parameters to perform affine transformation on the bounding box, which represents the position of the predicted license plate. The max functions \( v_3 \text{ and } v_6 \) are used to ensure that the diagonal is positive (avoid unnecessary mirroring or excessive rotation).

To match the output resolution of the network, the points \( m_i \) are re-scaled by the inverse of the network stride and are re-centered according to each point \( (i, j) \) in the feature map. This is performed by using a normalization function:

\[
A_{ij}(m) = \frac{1}{\beta} \left( \frac{1}{N_s} m - \begin{bmatrix} i \\ j \end{bmatrix} \right)
\]

(2)

where \( \beta \) is a scaling constant that denotes the side of the virtual square. We set \( \beta = 7.75 \), which is the mean point between the maximum and minimum dimensions of the candidate LPs in the augmented training data that are divided by the network stride. What needs to be added is that in order to maintain a good compromise between accuracy and processing time, we chose the maximum dimension of 608 and the minimum dimension of 288 according to the experiments.

Assuming that there is an object (LP) at point \( (i, j) \), the first part of the loss function considers the error between a distorted version of the canonical square and the normalized annotated points of the LP, given by

\[
f_{\text{affine}}(i, j) = \sum_{k=1}^{4} \left\| T_{ij}(n_k) - A_{ij}(m_k) \right\|
\]

(3)

The second part of the loss function handles the probability of having/not having an object at \((i, j)\). The loss is the sum of two log-loss functions:

\[
f_{\text{prob}}(i, j) = \log \text{loss}(\Pi_{\text{obj}}, v_1) + \log \text{loss}(1 - \Pi_{\text{obj}}, v_2)
\]

(4)

where \( \Pi_{\text{obj}} \) is the object indicator function that return 1 if there is an object at point \((i, j)\) or 0 otherwise.

The final loss function is given by a combination of the terms defined in Eqs. (3) and (4):

\[
\text{Loss} = \sum_{i=1}^{P} \sum_{j=1}^{Q} \left[ C_{\text{obj}} f_{\text{affine}}(i, j) + f_{\text{prob}}(i, j) \right]
\]

(5)

C. LICENSE PLATE RECOGNITION

We reconstructed a robust three-level LPR network framework. The three steps of the LPR module are shown in Figure 5.

1) FEATURE EXTRACTION (FEAT.)

At this stage, since the license plate distortion correction network has corrected the license plate, we can directly use CNN to perform feature extraction on the input image. We studied three architectures of VGG [58], RCNN [59], and ResNet [60]. VGG is originally composed of multiple convolutional layers and several fully connected layers. RCNN is a variant of CNN that can be recursively applied for the adjustment of its acceptance field according to the shape of characters. ResNet is a CNN with residual connections, which can alleviate the burden of relatively deep CNN training. Experimental tests show that ResNet has achieved the best experimental results under the same experimental conditions.

2) SEQUENCE MODELING (SEQ.)

At this stage, we focused on testing the effect of the BiLSTM [61]. Long Short-Term Memory (LSTM) is a type of RNN, and BiLSTM is the abbreviation of Bi-directional LSTM, which is composed of forwarding LSTM and backward LSTM. Unlike other LPs that only contain English characters and numbers, Chinese LPs also contain some Chinese characters. Chinese is in the first place, and the remaining English
characters and numbers are arranged behind Chinese, which makes it difficult to recognize. Based on the actual situation, we can choose to use BiLSTM or not.

3) PREDICTION (PRED.)
To summarize the previous work, we have two prediction methods: (1) Connectionist temporal classification (CTC) [62]; and (2) attention-based sequence prediction (Attn) [63]. The key method of CTC is to predict a character in each column and alter the entire character sequence into a non-fixed character stream by removing repeated characters and whitespace. The CTC method only considers the results of an input sequence to an output sequence. It only considers whether the predicted output sequence is related to the real sequence, instead of considering whether each result in the predictive output sequence is perfectly aligned with the input sequence. Unlike CTC, Attn automatically captures the information flow in the input sequence for the prediction of the output sequence. The attention-based model is a similarity measurement. The more similar the current input is to the object state, the greater the weight of the current input is, which indicates that the current output is more dependent on the current input. In this work, we chose to use an attention-based mechanism to automatically capture the information flow in the input sequence for the prediction of the output sequence.

The detection and recognition modules can fulfill end-to-end training without character segmentation, and it can recognize license plate characters of any length.

IV. EXPERIMENTS
In this chapter, we verified the effectiveness of the proposed method through experiments. Our network was fulfilled using Pytorch, and the experiments were carried out on the Intel Xeon E5-2680v4 processor, Nvidia Tesla K80×8 GPU.

A. DATASETS
We used the CCPD dataset for the model training. CCPD provides over 250k unique License plate images and detailed annotations. The resolution of each image is 720 (w)×1160 (h)×3 (channel). The samples were collected under different lighting and different weather conditions.

Although CCPD is diverse, the data is relatively uneven. In deep learning, the number of training samples is generally required to be sufficient. The larger the number of samples, the better the trained model and the stronger the generalization ability of the model. To make the model performance more powerful, it is essential to use data augmentation.

We generated about 2,700k LPs by the following augmentation methods in this paper:
1) Rectification: the entire image is rectified based on the LP annotation, assuming that the LP lies on a plane;
2) Centering: setting the center of the License plate as the image center;
3) Geometric transformations: scaling, translation, cropping, mirroring, and rotation;
4) Color space: slight modifications in the HSV color space.
We can obtain a variety of visually augmented images from a manually marked sample with the above transformations. For example, Figure 6 shows 8 different augmented samples obtained from the same images.

To verify the performance of the ALPR system designed in this paper, we collected 300 Chinese license plate images with different degrees of skewing and distortion as the test set. The data were collected under different lighting and different weather conditions. Compared with the mainstream training set, all the LPs in the test data set have a large angle of inclination or even a serious warp.

**B. TRADE-OFF ANALYSIS OF ALGORITHMS COMBINATION**

In the LPR module, we paid attention to the trade-off between speed and accuracy in different algorithm combinations. As shown in Table 1, “Acc” means the proportion of the correct license plates in the total number, and “Time” represents the average processing time of each image. We can learn that R1 has the simplest structure and the shortest time. From R1 to R7, our framework provides a smooth method conversion, and you can choose different solutions according to different scenarios. They in turn increase the complexity of the entire LPR model, thereby improving performance at the cost of computational efficiency. The introduction of ResNet and BiLSTM did not cause a significant drop in speed and greatly improved the accuracy (81.0% → 93.7%). From R2 to R3, the performance improvement after introducing BiLSTM is not very obvious (92.3% → 93.7%). From R4 to R6, we observed that the Seq and Pred modules did not contribute much to memory consumption (1.9M → 5.5M parameters). Although generally lightweight, these modules provided improved accuracy (87.7% to 94.3%). From R6 to R7, using ResNet increased accuracy by 4.8% at the cost of memory consumption increasing from 5.5 million to 47.9 million floating-point parameters. From R3 and R7, the introduction of the Attn mechanism improved the accuracy by 5.5%, but it took a relatively long time. Our ALPR system chose R7 in this work.

**C. ALPR SYSTEM RESULTS DISPLAY**

To see the power of the distortion correction network of the LPD module more intuitively, we conducted special tests accordingly. The experimental results are shown in Figure 7.
It can be seen from Figure 7 that in the case where the camera angle is severely skewed or even those LPs are deformed, our proposed method can still convert the skewed and distorted license plate images into rectangular ones taken from an almost frontal perspective. Among them, the red borders represent the positioning of the LPs by the algorithm. Our proposed method can accurately locate the LPs even at night, which confirms the high performance of the proposed method.

For better operability, we used PyQT to encapsulate the source program and displayed the detected vehicles and LPs on the right in real-time. The GUI interface is shown in Figure 8. For reasons of clarity, we only show the recognition results close to the camera. The video processing and playback are controlled by the following buttons:

1) **Load button:** Enter the path of the file to be displayed.
2) **Start button:** Start playing and processing the video.
3) **Pause button:** Pause the video playback to observe the results of LPD and LPR. At this time, you can use the arrow keys ‘→’ on the keyboard to skip to the next frame. Click the Start button again to continue playing and processing the video.

|    | Feat.   | Seq.   | Pred. | Ace % | Time s | params |
|----|---------|--------|-------|-------|--------|--------|
| R1 | VGG     | None   | CTC   | 81.0  | 0.043  | 5599072|
| R2 | ResNet  | None   | CTC   | 92.3  | 0.048  | 44313152|
| R3 | ResNet  | BiLSTM | CTC   | 93.7  | 0.051  | 47180864|
| R4 | RCNN    | None   | CTC   | 87.7  | 0.047  | 1909216 |
| R5 | RCNN    | None   | Attn  | 92.7  | 0.049  | 2969825 |
| R6 | RCNN    | BiLSTM | Attn  | 94.3  | 0.050  | 5534433 |
| R7 | ResNet  | BiLSTM | Attn  | 98.9  | 0.053  | 47938369|
As an additional contribution, we proposed a new challenging data set for evaluating the ability of the ALPR system to capture severely skewed and distorted LPs. The annotations for the data set will be publicly available so that the data set can be used as a new and challenging benchmark for ALPR research.

For our future research, we intend to explore a new CNN architecture to further optimize vehicle detection (in terms of speed). And we also plan to extend our solution to the detection of LPs for motorcycles and electric bicycles. Due to the different layout of license plates, this brings new challenges.

**REFERENCES**

[1] W. Wang, J. Yang, M. Chen, and P. Wang, “A light CNN for end-to-end car license plates detection and recognition,” IEEE Access, vol. 7, pp. 173875–173883, 2019.

[2] W. Weihong and T. Jiayang, “Research on license plate recognition algorithms based on deep learning in complex environment,” IEEE Access, vol. 8, pp. 91661–91675, 2020.

[3] 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.

[4] S. Zhang, G. Tang, Y. Liu, and H. Mao, “Robust license plate recognition with shared adversarial training network,” IEEE Access, vol. 8, pp. 697–705, 2020.

[5] J. Zhuang, S. Hou, Z. Wang, and Z.-J. Zha, “Towards human-level license plate recognition,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 306–321.

[6] 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.

[7] 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. (ECCV), vol. 9905, Oct. 2016, pp. 21–37.

[8] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02767. [Online]. Available: http://arxiv.org/abs/1804.02767

[9] M. Raman and H. Aggarwal, “Study and comparison of various image edge detection techniques,” Int. J. image Process., vol. 3, no. 1, pp. 1–11, 2009.

[10] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, “Segmentation-and-annotation-free license plate recognition with deep localization and failure identification,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 9, pp. 2351–2363, Sep. 2017.

[11] C. Henry, S. Y. Ahn, and S.-W. Lee, “Multinational license plate recognition using generalized character sequence detection,” IEEE Access, vol. 8, pp. 35185–35199, 2020.

[12] H. Seibel, S. Goldenstein, and A. Rocha, “Eyes on the target: Super-resolution and license-plate recognition in low-quality surveillance videos,” IEEE Access, vol. 5, pp. 20020–20035, 2017.

[13] P. Venkateswari, E. J. Steffy, and D. N. Muthukumar “License Plate cognizance by Ocular Character Perception,” Int. Res. J. Eng. Technol., vol. 5, no. 2, pp. 536–542, Feb. 2018.

[14] A. C. Roy, M. K. Hossen, and D. Nag, “License plate detection and character recognition system for commercial vehicles based on morphological approach and template matching,” in Proc. 3rd Int. Conf. Electr. Inf. Commun. Technol. (ICEEICT), Sep. 2016, pp. 1–6.

[15] R. Panahi and I. Gholampour, “Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 4, pp. 767–779, Apr. 2017.

[16] 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.

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

[18] R. C. Chen, “Automatic license plate recognition via sliding-window darknet-YOLO deep learning,” Image Vis. Comput., vol. 87, pp. 47–56, Jul. 2019.

---

**TABLE 2.** Comparison of two sets of training data.

| Methods    | Real % | Real + Augmentation % |
|------------|--------|------------------------|
| Li et al. [22] | 77.3   | 83.3                   |
| Li et al. [30] | 85.0   | 88.3                   |
| Yuan et al. [25] | 89.7   | 93.2                   |
| Laroca et al. [32] | 82.6   | 86.0                   |
| Xie et al. [34] | 89.6   | 94.6                   |
| Xu et al. [36] | 90.6   | 95.7                   |
| Ours        | 93.6   | 98.3                   |

4) **Bounding Box button:** The video display result will show the license plate recognition information.

**D. PERFORMANCE COMPARISON WITH OTHER ALPR SYSTEMS**

This part covers the experimental analysis of our entire ALPR system, as well as comparison with other advanced methods. Unfortunately, most academic ALPR papers only focus on simple scenarios (e.g., isolated environmental conditions, camera position, etc.). Besides, many papers only focus on License plate detection or character segmentation, which even limits the possibilities for comparison of the entire framework.

In this work, we use the different scene data sets we collected for the accuracy evaluation of the proposed method in different scenes. We evaluate the system on the basis of the percentage of correct identification. A License plate can be considered correct if all the characters can be correctly recognized. To demonstrate the benefits of using data augmentation during training, we use two sets of training data for evaluation: (i) real data plus artificially generated data; (ii) data in natural scenes only.

From Table 2 we can see that adding synthesized data improves the accuracy of all methods (gain ≈ 5%). Compared with other methods, our method has an absolute advantage. Most previous methods have only been tested on datasets with positive perspectives, but there will be considerable accuracy degradation in challenging oblique datasets. The large increase in accuracy in this work is mainly due to the distortion correction network that changed the inclined license plate in advance, and then cooperated with a powerful LPR network. Even in the case of severe skewing and distorting of the license plate, ALPR tasks can be accomplished excellently.

**V. CONCLUSION**

In this work, we proposed a real-time and robust Chinese ALPR method in natural scenes. It can quickly and accurately detect and identify inclined and twisted LPs. Many comparative experiments were conducted on the challenging data sets we collected. The experimental results show that the method is far superior to the existing ALPR methods in terms of performance.
MING-XIANG HE is currently a Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology, China. He is also a member of the National Virtual Simulation Experiment Center of Shandong University of Science and Technology. His current research interests include image processing, artificial intelligence, and database systems.

PENG HAO is currently pursuing the master’s degree in software engineering with the Shandong University of Science and Technology, China. His current research interests include deep learning, and object detection and text recognition.

***