A High-performance Approach for Irregular License Plate Recognition in Unconstrained Scenarios

Hoanh Nguyen
Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

***Abstract*—This paper proposes a novel framework for locating and recognizing irregular license plates in real-world complex scene images. In the proposed framework, an efficient deep convolutional neural network (CNN) structure specially designed for keypoint estimation is first employed to predict the corner points of license plates. Then, based on the predicted corner points, perspective transformation is performed to align the detected license plates. Finally, a lightweight deep CNN structure based on the YOLO detector is designed to predict license plate characters. The character recognition network can predict license plate characters without depending on license plate layouts (i.e., license plates of single-line or double-line text). Experiment results on CCPD and AOLP datasets demonstrate that the proposed method obtains better recognition accuracy compared with previous methods. The proposed model also achieves impressive inference speed and can be deployed in real-time applications.**

Keywords*—License plate recognition; deep learning; convolutional neural network; keypoint detector; YOLO detector

I. INTRODUCTION

Recognizing license plates is a crucial area of research due to its numerous practical applications, including monitoring road traffic, collecting tolls automatically, enforcing traffic laws, and more. A license plate recognition pipeline for recognizing irregular license plates typically includes four stages: license plate detection, perspective correction, segmenting characters and recognizing characters. License plate detection aims to extract license plate regions from input images. The accuracy of the entire system heavily relies on the accuracy of license plate detection, as the extracted regions are utilized in subsequent stages. As real-world images containing license plates are captured under different viewpoints, license plates may have arbitrary direction. As a result, perspective correction is performed to align the detected license plates. For recognizing characters, a segmentation approach is first used to decompose the aligned license plate image containing a sequence of characters into sub-images of individual character. Then, a character recognition approach is employed to classify each character. Classical approaches based on computer vision for license plate recognition [1], [2] primarily focus on extracting features of license plates based on the background color, contours and edges, and use these hand-crafted features for locating license plates and decomposing characters. Recently, numerous CNN-based approaches for license plate recognition have been proposed, leading to significant advancements. These methods first adopt CNN architectures to extract discriminative feature representations from input images. A network is then used to locate the location of license plates. With the detected license plates, a classifier is proposed to search for license plate characters and classify them. Since character segmentation is a challenge problem due to the effect of lighting conditions, shadows, and noise, various approaches [3], [4] have been proposed to directly recognize license plate characters without segmentation. With the success of CNN and text recognition, CNN-based license plate recognition methods have obtained great achievements in both accuracy and efficient. However, previous methods still depend on high-end GPUs or controlled environments such as specific viewing angles or simple backgrounds. Furthermore, with the growing number of license plate designs, license plate recognition systems that concentrate on single-line plates or frontal plate detection and recognition face increasing challenges.

In view of these issues, a novel framework for detecting and recognizing irregular license plates in real-world complex scene images is designed in this paper. The proposed model can locate and recognize various types of license plates with arbitrary shooting angles in difficult conditions. There are three stages in the proposed model: license plate detection, perspective correction, and recognizing characters. For license plate detection, this paper employs a state-of-the-art object detector and extends it for predicting four corner points of license plates, which are then used to rectify distorted license plates. For license plate recognition, this paper designs a segmentation-free model based on a fast and efficient object detection architecture for predicting license plate characters. The results of the experiments conducted on two extensive datasets show that the proposed model boasts both a high recognition accuracy and rapid inference speed.

This paper is structured as follows: Section II presents a literature review of license plate recognition. Section III offers an overview of the proposed approach. The details of the proposed pipeline are described in Section IV. The discussion of experimental results can be found in Section V. Lastly, the conclusion is outlined in Section VI.

II. LITERATURE REVIEW

This section provides a brief literature review on the topic of license plate recognition. This paper focuses on recent methods that are based on deep learning for end-to-end license plate recognition. For studies involving traditional image processing strategies or focused on license plate detection, please refer to [1], [2], [5], [30].

Since license plates usually occupy small portions of input images, various methods proposed to first detect vehicles and then locate license plate regions to improve license plate detection performance. For this purpose, Sergio and Claudio [6] proposed a novel CNN model that includes a YOLO-based network for vehicle detection and license plate detection and an optical character recognition module for character recognition. The model can detect and rectify multiple distorted license plates before feeding the rectified license plates to the optical character recognition module to obtain results. In [7], the authors presented an end-to-end license plate recognition system utilizing the YOLO detector [8], [9]. This approach first locates the vehicles in the input image by a YOLO detector and then detects their respective license plates in the vehicle patches by another YOLO detector. Afterward, the model detects and recognizes all license plate characters simultaneously by forwarding the license plate region into the CR-NET model [10]. The results showed that this approach obtains high accuracy and fast inference speed. However, the model only recognizes single-line license plate taken from a frontal angle.

Due to the effect of environment conditions, character segmentation is a challenge problem. Moreover, any incorrect character location produced by character segmentation will lead to misrecognition of the license plate characters. To solve this problem, various methods proposed to avoid character segmentation. Wang et al. [11] introduced a cascade approach (i.e., VSNet) for irregular license plate recognition. VSNet consists of a license plate detection network that makes predictions using multiple feature levels produced by a fusion network and a license plate recognition network that features an encoding layer for left-to-right feature extraction and a weight-sharing classifier for character recognition. In addition, a vertex-estimation branch is proposed to rectify distorted license plate images. In [12], the authors presented an end-to-end convolutional neural network for license plate recognition that eliminates the need for character segmentation. The network is implemented on FPGA with very fast processing speed. To enhance the accuracy of license plate recognition in unrestricted conditions, Zou et al. [13] proposed a robust license plate recognition framework that uses a combination of Bi-LSTM and contextual position information of license plate characters to locate the characters in the license plate. The authors utilized deep separable convolutions and a spatial attention mechanism for license plate feature extraction to activate the character feature regions and thoroughly extract the features of license plates.

In summary, although the above methods have achieved some significant accomplishments, they have not fully addressed the issue of irregular license plate recognition in unconstrained scenarios. Furthermore, these methods mostly require hardware with high-end GPUs, which is difficult to implement in practical applications.

III. Overview of the Proposed Framework

The proposed method includes three stages as outlined in Fig. 1. Specifically, the proposed method takes images as inputs and sequentially undergoes license plate detection, perspective correction, and character recognition to produce final license plate characters. Both stage 1 and stage 3 are based on simple and efficient deep CNN architectures for fast inference speed. Overview of each stage is described further.

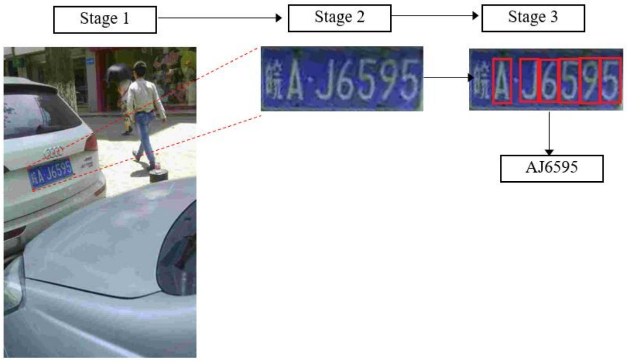

Fig. 1. Overview of the proposed framework.

Stage 1: License plate detection. As shown in Fig. 1, license plate detection aims to locate the four corner points of each license plate. For this purpose, this paper employs a lightweight deep CNN structure used for human pose estimation [14] and modifies it for predicting the four corner points of license plates.

Stage 2: Perspective correction. Perspective deformation images are corrected by applying perspective transformation. First, four corner points are predicted by the license plate detection network. Then, the homography between the camera and the license plate is recovered. Finally, the homography is used to warp the detected license plate into a rectified image as shown in Fig. 1.

Stage 3: Character recognition. For reading license plate characters, this paper considers character recognition as character detection problem and designs a lightweight character detection network that predicts all license plate characters on the detected license plates. The character detection network can predict license plate characters without depending on license plate layouts.

IV. Implementation Details

Fig. 2 illustrates the detailed pipeline of the proposed model. Input images are fed into a fast and efficient license plate detection network for predicting the four corner points of license plate (i.e., top-left, bottom-left, top-right, and bottom-right corners). Based on the detected corner points, perspective transformation is performed to align the detected license plate. Finally, a lightweight deep CNN network is employed to predict license plate characters. The following subsections illustrates the details of each stage.

A. License Plate Detection

This paper uses CenterNet [14] for extracting license plate regions and the corresponding corner points from input images. CenterNet considers the center point of a bounding box as an instance and uses this keypoint to predict the dimensions and offsets of the box. CenterNet strikes a desirable balance between precision and speed and is highly customizable and extensible. It can be easily extended to multiple computer vision tasks including 3D object detection, object tracking, human pose estimation, and many others. In this paper, CenterNet is used to predict the four corner points of license plates (i.e., top-left, bottom-left, top-right and bottom-right corners). Based on the predicted corner points, perspective correction is performed to get rectified license plate images. For this purpose, this paper employs the CenterNet structure used for human pose estimation [14] and modifies it for corner point estimation. The detailed pipeline of the license detection model based on CenterNet is shown in Fig. 3. Consider an input image $I \in \mathbb{R}^{H\times W\times 3}$, where W and H represent the width and height of the input image, respectively, a fully convolutional encoder-decoder architecture is first used to produce feature representations from input images and generate output results. Three heads are produced after one forward pass from the feature extraction network as shown in Fig. 3 (i.e., keypoint heatmap head, corner location head, and corner offset head). All the heads are predicted with the same dimensions (i.e., height and width) $(H/s, W/s)$, where s represents the output stride ($s = 4$ in this paper).

1) Feature extractor. This paper adopts RestNet-50 [15] for feature extraction. ResNet blocks are then augmented with three up-convolutional layers to incorporate higher resolution output feature maps. In addition, a 3×3 deformable convolutional layer is used before each up-sampling layer.

2) Keypoint heatmap head. Keypoint heatmap head is used for predicting the center point of license plates. In this paper, the keypoint heatmap $\hat{K} \in \mathbb{R}^{\frac{H}{S}\times\frac{W}{S}\times 1}$ has one channel since only one class is predicted by the license plate detection network. After one forward pass, a Sigmoid layer is utilized on the keypoint heatmap, and the calculated value at each keypoint is viewed as the certainty score for it being the center of the license plate.

3) Corner locations head. Corner location head predicts the four corner locations of license plate (i.e., top-left, bottom-left, top-right, and bottom-right corners). Each corner is considered as a 2-dimensional property of the center keypoint and parameterized by an offset to the center keypoint. The dimensions of this head are $(\frac{H}{S}\times\frac{W}{S}\times 8)$.

4) Corner offset head. The Corner offset head is employed to rectify the quantization error resulting from the down-sampling of the input. After one forward pass, the coordinates of predicted center keypoints are mapped to a higher resolution input image. This results in a deviation in values because the original image coordinates are whole numbers, whereas the actual center points ought to be decimal numbers. As a result, the local offsets $\hat{O} \in \mathbb{R}^{\frac{H}{S}\times\frac{W}{S}\times 2}$ are predicted for each center point to recover the discretization error.

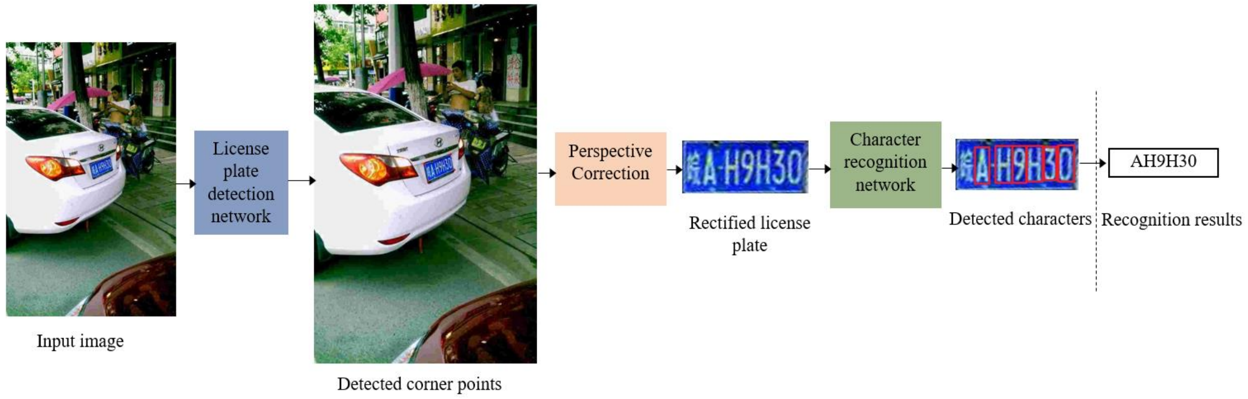

Fig. 2. The detailed structure of the proposed method.

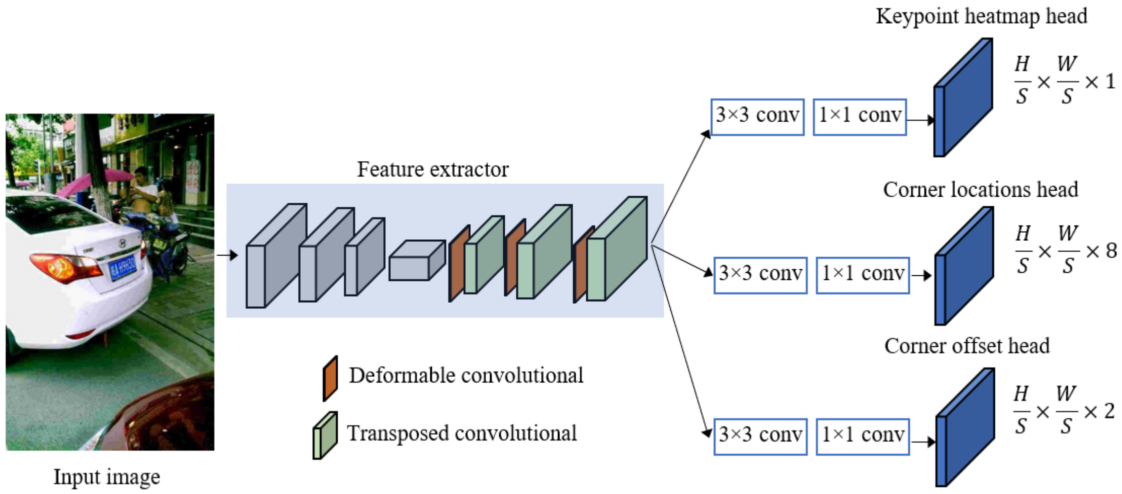

Fig. 3. The architecture of the license plate detection network.

B. Perspective Correction

As license plates can sometimes be difficult to read due to the viewpoint, perspective correction is performed to align the detected license plates. Based on the four corner points generated by the license plate detection network, the homography between the camera and the license plate is first recovered. Then, the homography is used to warp the detected license plate into a rectified image as shown in Fig. 1. To be more specific, based on the detected corner points from input image, this paper first identifies x_{max} and y_{max} , which represent the maximum horizontal and vertical distances between the corner points. Then four corresponding vertices of rectified image are calculated as follow:

$$(tl, tr) = ((0,0), (x_{max} - 1,0)) \quad (1)$$

$$(br, bl) = ((x_{max} - 1, y_{max} - 1), (y_{max} - 1,0)) \quad (2)$$

where tl, tr, br, bl represent the top-left, top-right, bottom-left, and bottom-right corners of the rectified license plate.

Following [16], the perspective transformation matrix H is calculated from the detected corner points and corresponding vertices as follow:

$$\begin{bmatrix} t_i x_i' \\ t_i y_i' \\ t_i \end{bmatrix} = H. \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (3)$$

where:

$$dst(i) = (x_i', y_i');\ src(i) = (x_i, y_i), i = 0, 1, 2, 3 \quad (4)$$

and

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \quad (5)$$

Finally, the rectified license plate region is formed as follow:

$$dst(x, y) = src\left(\frac{H_{11}x + H_{12}y + H_{13}}{H_{31}x + H_{32}y + H_{33}}, \frac{H_{21}x + H_{22}y + H_{23}}{H_{31}x + H_{32}y + H_{33}}\right) \quad (6)$$

C. Character Recognition

Character recognition aims to identify each character on the rectified license plates. For this purpose, this paper considers character recognition as character detection problem and designs a lightweight character detection network that predicts each license plate character without depending on license plate layouts (i.e., license plates of single-line or double-line text). Specifically, the lightweight character detection network is trained to detect 35 classes (i.e., ‘A-Z’, ’0-9’, the digit ‘0’ is recognized jointly with the letter ‘O’) based on the rectified license plates as well as the bounding box and class of each character as inputs. In the case of Chinese license plates, the initial symbol is a Chinese character that signifies the province. As stated in [17] and [7], the character detection network proposed in this work has not been trained to identify Chinese characters because assigning the category to such characters is not a straightforward task.

Table I showcases the design of the suggested lightweight character detection network. The design of the network is influenced by the Fast-YOLOv4 model [18], which is a tiny deep neural architecture that obtains very fast detection speed without sacrificing much accuracy. As shown in Table I, 3×3 convolution layers are used to extract features from previous layers followed by 1×1 convolution layers for reducing the feature channels. In addition, max pooling layers are used to decrease the feature dimensions. The number of channels is multiplied by two following each max pooling layer. Following [9], [19], this paper applies detection head at different scales to predict license plate characters. Specifically, character prediction is performed at layer 13 and layer 20, where the output size is 24×8 and 48×16, respectively. This detection approach is crucial for character recognition because the characters on the license plate may take up either a small or large area of the license plate region, as depicted in Fig. 4. It is worth mentioning that the proposed license plate character recognition system accurately detects and identifies license plates with either single-line or double-line text as it predicts all characters on the rectified license plate.

TABLE I. THE STRUCTURE OF THE PROPOSED LIGHTWEIGHT CHARACTER DETECTION NETWORK

Layer	Type	Filter size, stride	Input size	Output size
0	Conv	3×3×32,1	384×128×3	384×128×32
1	Max-pool	2×2, 2	384×128×32	192×64×32
2	Conv	3×3×64,1	192×64×32	192×64×64
3	Max-pool	2×2, 2	192×64×64	96×32×64
4	Conv	3×3×128,1	96×32×64	96×32×128
5	Max-pool	2×2, 2	96×32×128	48×16×128
6	Conv	3×3×256,1	48×16×128	48×16×256
7	Conv	1×1×128,1	48×16×256	48×16×128
8	Conv	3×3×256,1	48×16×128	48×16×256
9	Max-pool	2×2, 2	48×16×256	24×8×256
10	Conv	3×3×512,1	24×8×256	24×8×512
11	Conv	1×1×256,1	24×8×512	24×8×256
12	Conv	3×3×512,1	24×8×256	24×8×512
13	Conv	1×1×45,1	24×8×512	24×8×45
14	Detection			
15	Route [11]			24×8×256
16	Conv	1×1×256,1	24×8×256	24×8×256
17	Upsample	2×	24×8×256	48×16×256
18	Route [17, 6]			48×16×512
19	Conv	3×3×512,1	48×16×512	48×16×512
20	Conv	1×1×45,1	48×16×512	48×16×45
21	Detection			

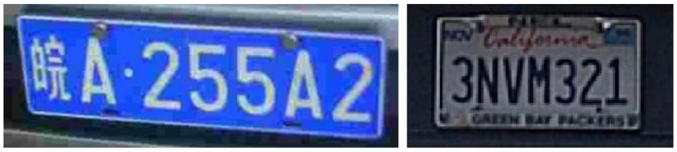

Fig. 4. License plate characters may occupy either small (Right) or large portion (Left) on the license plate region depending on type of license plate.

V. Experimental Results

All experiments were carried out on a computer equipped with an Intel Core i7-10700 CPU, a single NVIDIA GeForce GTX 1080Ti GPU, and 64GB of RAM. All models are designed and evaluated under the framework of PyTorch [20] and mmdetection [21].

A. Dataset and Evaluation Metrics

To assess the proposed method, this paper evaluates experiments on two extensive public datasets: CCPD [22] and AOLP [23].

CCPD [22] consists of 290k images captured under diverse illuminations, environments, and weather conditions. This dataset is more challenge than other datasets for license plate recognition since each image is captured from different positions and angles, which makes license plates have arbitrary direction. The dataset provides sufficient annotations for training the proposed model, including bounding boxes, vertices of each license plate, and license plate characters. Images in the dataset have the resolution of 720 (width) × 1160 (height) × 3 (channels). Following [22], this paper employs 100k images of CCPD-Base subset for training both detection and character recognition network. The remaining 100k images from the CCPD-Base subset and the 80k images from the CCPD-DB, CCPD-FN, CCPD-Rotate, CCPD-Tilt, CCPD-Weather, and CCPD-Challenge subsets are utilized for testing. Additionally, the CCPD dataset also includes the CCPD-Characters subset, consisting of over 1000 individual images for every possible license plate character. This paper utilizes the CCPD-Characters subset for further training the license plate recognition network.

AOLP (Application-Oriented License Plate database) [23] contains 2049 images. The images in this dataset are classified into three categories based on the capturing conditions: access control (AC), traffic law enforcement (LE), and road patrol (RP). The AC subset consists of 681 images of license plates captured in scenarios where vehicles move through a fixed passage at a slower pace or come to a complete stop. The LE subset consists of 757 images of license plates captured by a roadside camera during instances of traffic law violations. The RP subset includes 611 images of license plates captured from vehicles with random viewpoints and distances, making it more challenging for license plate recognition due to the heavily distorted license plates. Since the AOLP dataset only provides annotations for the coordinates of license plate bounding boxes and numbers, this paper manually annotates the four corners of each license plate. In line with [11], this paper trains on the LE and RP subsets and uses the RP subset for testing.

For the evaluation metric, this paper calculates the accuracy of license plate recognition by dividing the number of correctly recognized license plates by the total number of license plates in the test set. A recognition is considered correct only if the IoU_{poly} is greater than 0.5 and all characters have been correctly recognized. Here, IoU_{poly} is calculated as follow:

$$IoU_{poly} = \frac{DP \cap GP}{DP \cup GP} \quad (7)$$

where DP is the detected polygon of license plate and GP is the ground truth polygon of the license plate. DP and GP are calculated based on the detected corner points and ground truth corner points, respectively.

B. Results on CCPD

Table II provides recognition results of the proposed approach and recent approaches on the CCPD dataset. The results demonstrate that the proposed approach obtains the best recognition accuracy on most of the subsets. To be more specific, the proposed model obtains recognition accuracy at 99.7%, 99.2%, 99.1%, 99.6%, and 99.6% on CCPD-Base, CCPD- DB, CCPD- FN, CCPD- Rotate, and CCPD- Tilt, respectively, which outperform all previous methods, including method proposed by Zhang et al. [3]. For CCPD- Challenge subset, method proposed by Zhang et al. [3] obtains the best recognition accuracy. Since CCPD-Challenge subset comprises the most difficult images for the recognition of license plates, the simple and efficient license plate detection network cannot locate some license plates (Fig. 6), which leads to wrong recognition results by the character recognition network. In the future, this paper will investigate more effective fusion strategies to enhance the feature representation of the license plate detection network, which would improve detection results. It is noteworthy that the majority of the comparison methods in Table II determine the recognition results by setting the IoU threshold to 0.6. Additionally, it is observable that the proposed method attains the most significant improvements in the CCPD-Rotate and CCPD-Tilt subsets. Specifically, the proposed network improves recognition accuracy by 3.2% and 2% on CCPD-Rotate and CCPD-Tilt subsets compared with that of model proposed by Zhang et al. [3]. Given that the CCPD-Rotate and CCPD-Tilt subsets contain images with significant perspective distortion, these results show that the proposed model excels at detecting and recognizing license plates that have undergone distortion or rotation. For recognition speed, since the proposed model is designed based on fast and efficient architectures, it obtains the fastest recognition speed among comparing methods. To be more specific, the proposed model needs 8.3ms for processing an image based on single NVIDIA GeForce GTX 1080Ti GPU. The results indicate that the proposed model is efficient and well-suited for real-time applications. As depicted in Fig. 5, this study showcases the recognition results of the proposed approach on the CCPD dataset, including the detection of the four corners of the license plates, the rectified license plates after perspective correction, and the recognition of the characters on the license plates. It is evident that the proposed model performs effectively under various conditions. Fig. 6 shows some failure cases where the proposed model cannot locate license plates in challenging environments or fails to recognize some similar license plate characters.

TABLE II. THE RECOGNITION RESULTS OF THE PROPOSED MODEL AND OTHER MODELS AS EVALUATED ON THE CCPD DATASET

Method	Recognition Accuracy (%)							Inference Time (ms)
	Base	DB	FN	Rotate	Tilt	Weather	Challenge	
SSD [24] + HC [25]	98.3	96.6	95.9	88.4	91.5	87.3	83.8	25.6
YOLO9000 [8] + HC [25]	98.1	96.0	88.2	84.5	88.5	87.0	80.5	23.8
Faster-RCNN [26] + HC [25]	97.2	94.4	90.9	82.9	87.3	85.5	76.3	57.6
TE2E [17]	97.8	94.8	94.5	87.9	92.1	86.8	81.2	310
LPRNet [27]	97.8	92.2	91.9	79.4	85.8	92.0	69.8	17.8
RPnet [22]	98.5	96.9	94.3	90.8	92.5	87.9	85.1	11.7
Zou et al. [13]	99.3	98.5	98.6	92.5	96.4	99.3	86.6	-
DAN [28]	98.9	96.1	96.4	91.9	93.7	95.4	83.1	19.3
Zhang et al. [3]	99.6	98.8	98.8	96.4	97.6	**98.5**	**88.9**	24.9
Proposed method	**99.7**	**99.2**	**99.1**	**99.6**	**99.6**	**98.5**	88.5	8.3

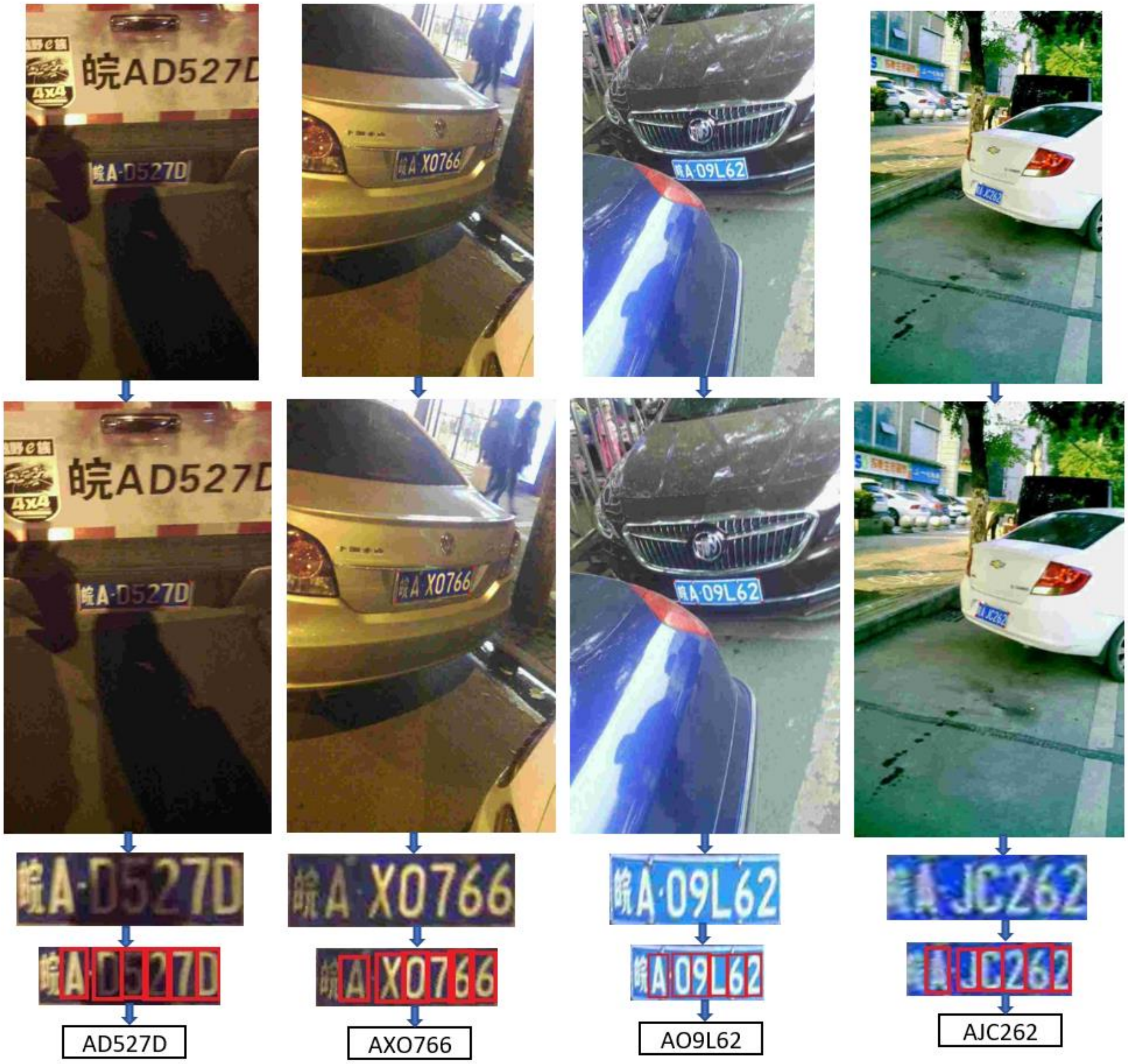

Fig. 5. Visual representation of the recognition outcomes of the proposed model on the CCPD dataset.

TABLE III. COMPARISONS OF RECOGNITION ACCURACY ON THE AOLP DATASET

Method	Recognition Accuracy (%)
Li et al. [29]	88.4
TE2E [17]	83.6
Zhang et al. [3]	91.9
Sergio et al. [6]	98.4
Zou et al. [13]	93.4
Proposed method	**98.8**

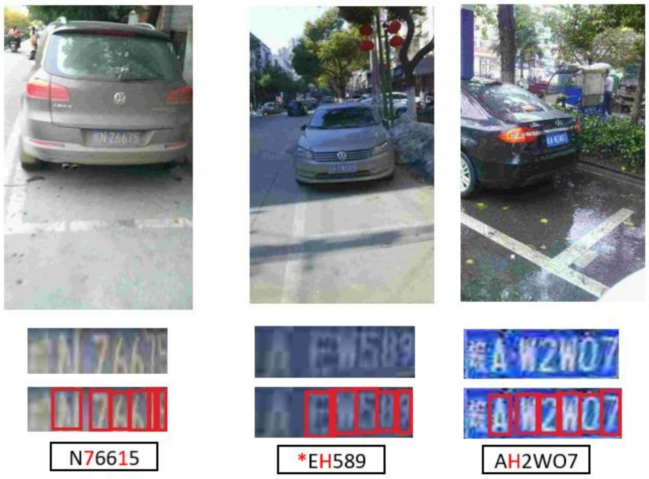

Fig. 6. Some examples of failure cases.

C. Results on AOLP

For the AOLP dataset, AOLP-RP subset is employed to evaluate the proposed model since this subset is more challenging (most of images contain severe perspective deformation license plates). Table III presents the comparison of recognition accuracy on the AOLP dataset. The proposed model emerges as the top performer in terms of recognition accuracy on the AOLP dataset, outperforming other methods. Specifically, the recognition accuracy of the proposed model surpasses that of the method proposed by Sergio et al [6] by 0.4%. This result further strengthens the claim of the proposed model's capability in recognizing license plates that have an irregular shape.

VI. CONCLUSION

This study presents a CNN-based approach for detecting and recognizing license plates with irregular shapes in complex real-world images. The proposed model employs a CenterNet-based CNN structure to predict the four corners of the license plates, followed by perspective correction to align the detected license plates. For character recognition, a YOLO-based segmentation-free model is designed to predict the characters on the license plate. The effectiveness of the proposed method is verified through experiments on the CCPD and AOLP datasets. Specifically, experimental results on two datasets show that the proposed method obtains the best recognition accuracy with the fastest recognition speed. This result demonstrates that the proposed method is highly suitable for intelligent traffic management applications that require real-time processing. In the future, the study intends to investigate additional fusion techniques for extracting more discriminative features from the input images, which can enhance the accuracy of the license plate detection network.

REFERENCES

[1] Anagnostopoulos, Christos-Nikolaos E., Ioannis E. Anagnostopoulos, Ioannis D. Psoroulas, Vassili Loumos, and Eleftherios Kayafas. "License plate recognition from still images and video sequences: A survey." *IEEE Transactions on intelligent transportation systems* 9, no. 3 (2008): 377-391.

[2] Du, Shan, Mahmoud Ibrahim, Mohamed Shehata, and Wael Badawy. "Automatic license plate recognition (ALPR): A state-of-the-art review." *IEEE Transactions on circuits and systems for video technology* 23, no. 2 (2012): 311-325.

[3] Zhang, Linjiang, Peng Wang, Hui Li, Zhen Li, Chunhua Shen, and Yanning Zhang. "A robust attentional framework for license plate recognition in the wild." *IEEE Transactions on Intelligent Transportation Systems* 22, no. 11 (2020): 6967-6976.

[4] Chen, Song-Lu, Chun Yang, Jia-Wei Ma, Feng Chen, and Xu-Cheng Yin. "Simultaneous end-to-end vehicle and license plate detection with multi-branch attention neural network." *IEEE Transactions on Intelligent Transportation Systems* 21, no. 9 (2019): 3686-3695.

[5] Nguyen, Hoanh. "Predicted anchor region proposal with balanced feature pyramid for license plate detection in traffic scene images." *Complexity* 2020 (2020).

[6] Silva, Sergio Montazzolli, and Claudio Rosito Jung. "License plate detection and recognition in unconstrained scenarios." In *Proceedings of the European conference on computer vision (ECCV)*, pp. 580-596. 2018.

[7] Laroca, Rayson, Luiz A. Zanlorensi, Gabriel R. Gonçalves, Eduardo Todt, William Robson Schwartz, and David Menotti. "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector." *IET Intelligent Transport Systems* 15, no. 4 (2021): 483-503.

[8] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263-7271. 2017.

[9] Redmon, Joseph and Ali Farhadi. "YOLOv3: An Incremental Improvement." *ArXiv* abs/1804.02767 (2018).

[10] Montazzolli, Sérgio, and Claudio Jung. "Real-time brazilian license plate detection and recognition using deep convolutional neural networks." In *2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pp. 55-62. IEEE, 2017.

[11] Wang, Yi, Zhen-Peng Bian, Yunhao Zhou, and Lap-Pui Chau. "Rethinking and designing a high-performing automatic license plate recognition approach." *IEEE Transactions on Intelligent Transportation Systems* (2021).

[12] Wang, Zhichao, Yu Jiang, Jiaxin Liu, Siyu Gong, Jian Yao, and Feng Jiang. "Research and Implementation of Fast-LPRNet Algorithm for License Plate Recognition." *Journal of Electrical and Computer Engineering* 2021 (2021).

[13] Zou, Yongjie, Yongjun Zhang, Jun Yan, Xiaoxu Jiang, Tengjie Huang, Haisheng Fan, and Zhongwei Cui. "A robust license plate recognition model based on bi-lstm." *IEEE Access* 8 (2020): 211630-211641.

[14] Zhou, Xingyi, Dequan Wang, and Philipp Krähenbühl. "Objects as points." *arXiv preprint arXiv:1904.07850* (2019).

[15] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[16] Andrew, Alex M. "Multiple view geometry in computer vision." *Kybernetes* (2001).

[17] Li, Hui, Peng Wang, and Chunhua Shen. "Toward end-to-end car license plate detection and recognition with deep neural networks." *IEEE Transactions on Intelligent Transportation Systems* 20, no. 3 (2018): 1126-1136.

[18] A. Bochkovskiy, “Fast-YOLOv4,” https://github.com/AlexeyAB/darknet/blob/master/cfg/yolov4-tiny.cfg, accessed: 2021-03-03.

[19] Bochkovskiy, Alexey, Chien-Yao Wang and Hong-Yuan Mark Liao. “YOLOv4: Optimal Speed and Accuracy of Object Detection.” ArXiv abs/2004.10934 (2020).

[20] Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in pytorch." (2017).

[21] Chen, Kai, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun et al. "MMDetection: Open mmlab detection toolbox and benchmark." *arXiv preprint arXiv:1906.07155* (2019).

[22] Xu, Zhenbo, Wei Yang, Ajin Meng, Nanxue Lu, Huan Huang, Changchun Ying, and Liusheng Huang. "Towards end-to-end license plate detection and recognition: A large dataset and baseline." In *Proceedings of the European conference on computer vision (ECCV)*, pp. 255-271. 2018.

[23] Hsu, Gee-Sern, Jiun-Chang Chen, and Yu-Zu Chung. "Application-oriented license plate recognition." *IEEE transactions on vehicular technology* 62, no. 2 (2012): 552-561.

[24] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In *European conference on computer vision*, pp. 21-37. Springer, Cham, 2016.

[25] Špaňhel, Jakub, Jakub Sochor, Roman Juránek, Adam Herout, Lukáš Maršík, and Pavel Zemčík. "Holistic recognition of low quality license plates by CNN using track annotated data." In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1-6. IEEE, 2017.

[26] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).

[27] Zherzdev, Sergey and Alexey Gruzdev. “LPRNet: License Plate Recognition via Deep Neural Networks.” *ArXiv* abs/1806.10447 (2018).

[28] Wang, Tianwei, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. "Decoupled attention network for text recognition." In *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, pp. 12216-12224. 2020.

[29] Li, Hui and Chunhua Shen. “Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs.” *ArXiv* abs/1601.05610 (2016).

[30] Nguyen, Hoanh. "An Efficient License Plate Detection Approach Using Lightweight Deep Convolutional Neural Networks." *Advances in Multimedia* 2022 (2022).