

Research on License Plate Detection and Recognition System based on YOLOv7 and LPRNet

Shenghu Pan, Jian Liu and Dekun Chen

Southwest Petroleum University, Chengdu 610500, China

Abstract: With the development and continuous iteration of digital transformation and artificial intelligence technology, the license plate detection and recognition system based on traditional machine vision and the current deep learning-based license plate recognition system for China is unable to meet the needs of rapid and accurate real-time recognition and recognition in complex environments. This paper designs and integrates a set of license plate detection and recognition system based on YOLOv7, STN and LPRNet models, which can recognize Chinese license plates quickly and accurately in real time, and has good robustness in complex environment. Its average accuracy in complex environments reached 96.1%, indicating that the system has a better effect than the traditional license plate detection and recognition system.

Keywords: Artificial intelligence, License plate detection, License plate recognition.

1. Introduction

In recent years, with the continuous development of artificial intelligence technology, computer vision has gained new development, has been applied to intelligent parking lot, intelligent security system and other work scenes, help to achieve intelligent urban security. License plate recognition system is one of the important applications of computer vision, it can automatically locate, detect license plates and recognize license plates. It has made great contributions to the construction of a modern and digital society [1]. However, with the development of society and economy, the number of vehicles in the society has a growing trend, and license plate recognition will be affected by complex environment, such as dark, long-distance, fog, etc. , therefore, there is a high demand for the speed, accuracy and ability to work in complex environment of license plate location detection and recognition.

In this paper, a new license plate detection and recognition system will be improved for the deficiencies of the current license plate detection and recognition system. First, YOLOv7 as one-stage object detection algorithm is used to quickly and accurately locate the license plate, and then the detected license plate is input into the STN (Spatial Transformer Networks) to correct the slanting license plate. And then input into the LPRNet to identify the license plate number. Finally, a set of license plate detection and recognition system based on PyQt5 is developed by integrating the above network structure. During the experiment, in order to improve the stability of the system, partial CCPD (Chinese City Parking Dataset) dataset, self-

collected dataset and mock dataset were used to realize uniform distribution of license plates in provinces and cities.

2. Main Work

2.1. Detection of License plate based on YOLOv7

YOLOv7 (you only look once) algorithm is a kind of real-time object detector proposed by Wang et al in July 2022. The essence of this algorithm is a regression problem, which has better effect when predicting multiple categories [2]. YOLO series algorithm, as a one-stage detection algorithm with simple structure and high computational efficiency, has become the mainstream algorithm for object detection at present. However, when applied to license plate location detection, its detection speed and accuracy cannot fully meet the requirements [3, 4]. The appearance of YOLOv7 makes up for these shortcomings to a large extent. Compared with the previous object detector, its biggest feature is that the prediction speed and accuracy have been greatly improved, and the real-time effect of the picture can be achieved when more than 30 FPS, with the highest AP accuracy reaching 56.8%. Therefore, this paper attempts to apply YOLOv7 to the end-to-end location detection of license plate, so as to improve the speed and accuracy of license plate location detection.

2.1.1. Network structure

The network structure of YOLOv7 is composed of input, backbone, feature pyramid network and detection head, as shown in **Figure 1**.

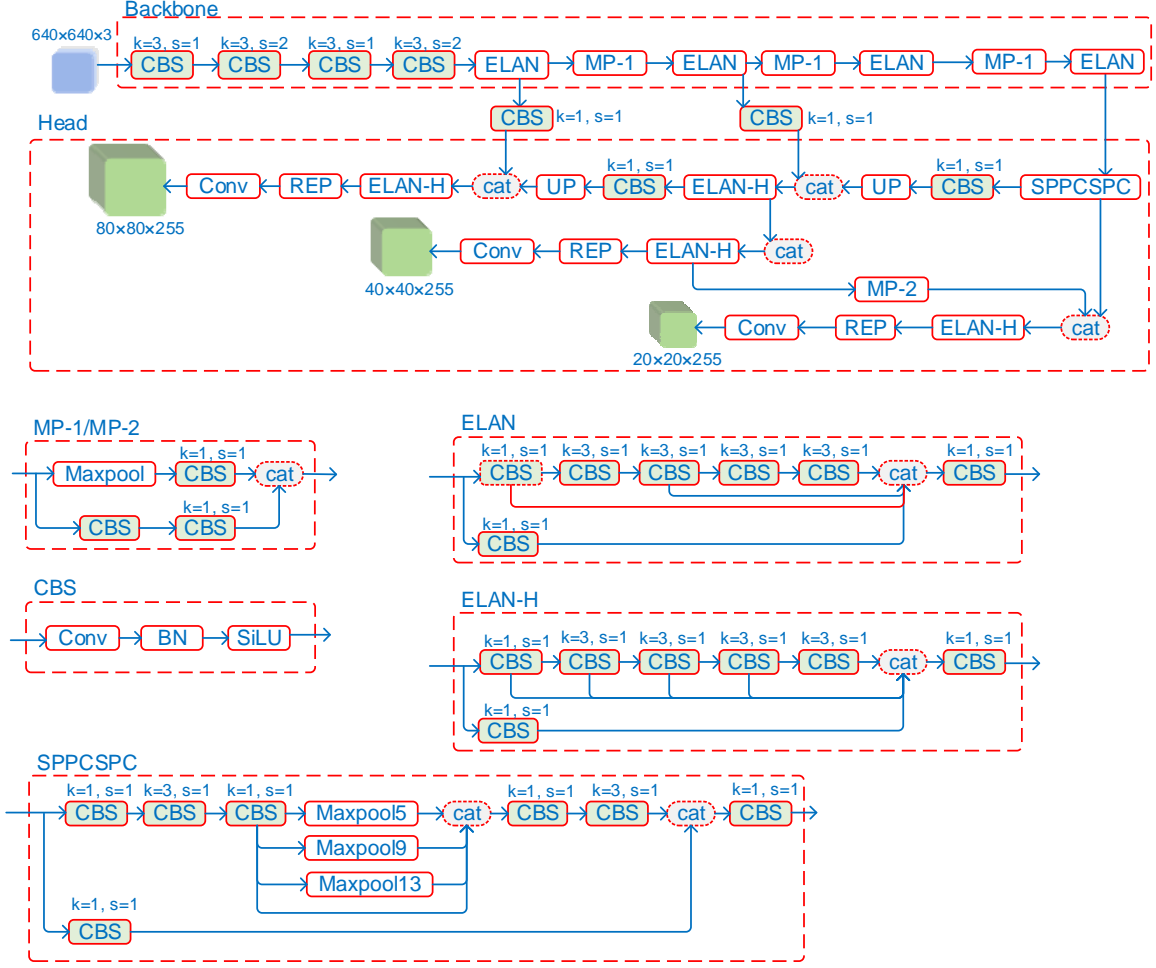


Figure 1. YOLOv7 network architecture

Firstly, the vehicle picture containing the license plate (size 640×640) is input to the backbone for feature extraction. The CBS module has different convolution cores and step sizes. The convolution of $k=1$ and $s=1$ is mainly used to change the number of channels; the convolution of $k=3$ and $s=1$ is mainly used for feature extraction; the convolution of $k=3$ and $s=2$ is mainly used for downsampling. After extraction, three feature maps of feature layers are output (size: 80×80×512, 40×40×1024, 20×20×1024, respectively). Then, the extracted three feature layers are fused in the feature pyramid network, so as to combine the feature information of different scales. SPPCSPC module is used to obtain different sensitivity fields through max pooling, which can better distinguish the big object from the small object. Secondly, the network adopts the efficient converged network ELAN module to control the shortest and longest gradient path to continuously enhance the ability of network learning, learn more features and have stronger robustness. Finally, the feature layer is enhanced to detect network modules and enter the model reparameterization module. Model reparameterization technology is a popular research direction at present. This technology divides the module into two parts: training and detection. During training, the module is divided into several branch structures, which can improve the number of features obtained by the network and achieve higher accuracy. During detection, multi-branch structures are integrated to form a completely equivalent module, which improves the speed of network detection.

After data processing in this paper, the training strategy with auxiliary head is used, which only appears in the training process. In the training, the training cost will be increased to improve the accuracy, but the detection time will not be affected.

2.2. License plate recognition based on LPRNet

License plate recognition is a complex and important task in urban traffic management, video surveillance, vehicle recognition and parking lot management. License plate recognition methods can be divided into two types: character segmentation and character non-segmentation. The method based on character segmentation usually uses edge detection, template matching and other methods to segment a single character in the license plate image, and then uses CNN (Convolutional Neural Network) and other CNNs to classify the characters. However, this method is greatly affected by external conditions such as illumination and is prone to errors. Zherzdev et al proposed LPRNet in 2018 [5]. The lightweight CNN structure is selected in this network, which can realize end-to-end license plate character recognition with character non-segmentation. The algorithm can be used to recognize license plate images under fuzzy, deformed, bad weather and other conditions. This section based on this to achieve the license plate recognition algorithm.

2.2.1. Network structure

LPRNet is a lightweight character recognition network, which is designed for the license plate scene with small character tag set and short string length. It has the

characteristics of high precision, few parameters and fast reasoning speed, and is suitable for embedded device deployment. The step of LPRNet algorithm is shown in **Figure 2**. [5, 6, 7]

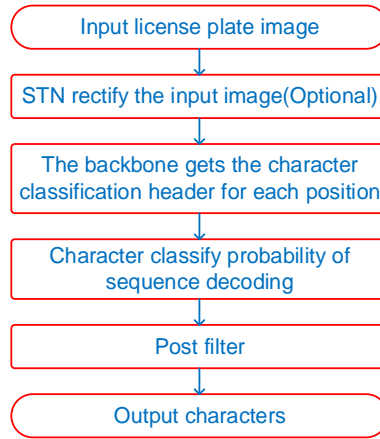


Figure 2. Steps of LPRNet algorithm

LPRNet uses STN at the input end to process input images. The STN architecture is shown in Figure 3 [8]. STN processing is an optional process, the appearance of STN enables us to complete how to transform the image to obtain excellent license plate recognition input image. After input, the image is sent to two convolution layers after 3x3 average

pooling, convolution is performed, and the two convolution feature maps are connected through concatenation operation to obtain features of different sizes. The effect is improved by preventing overfitting of the convolution layers in 50% dropout. Finally, it is output to the next module through two full connection layers.

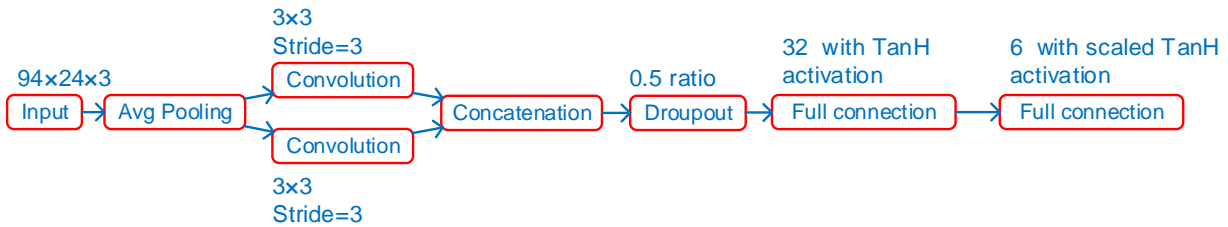


Figure 3. STN architecture

The backbone of LPRNet uses CNN to extract image features. The backbone inputs the native RGB image, and the backbone has 3 convolution layers, 3 pooling layers and 3 small basic block modules. Each small basic block module contains 4 convolution layers, so there are altogether 15 convolution layers and 3 pooling layers. The optimization method of Dropout [9] is used for regularization to prevent network overfitting. The output of the backbone can be thought of as a sequence representing the probability of the corresponding character, whose length is related to the width

of the input image. The last layer of the backbone uses a 1x13 convolution kernel, and the output result is expressed as a sequence of character probabilities, the length of which is associated with the width of the input Feature Map. To make the depth of the feature map equal to the number of character classes, a 1x1 convolution is used in the small basic block module to adjust the dimension. The backbone architecture is shown in **Figure 4**. The small basic block module is shown in **Figure 5**.

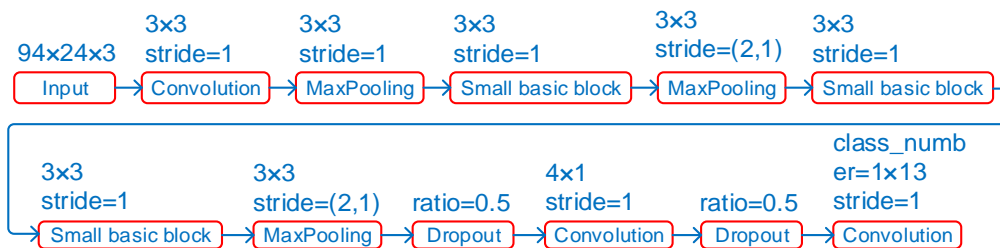


Figure 4. Back-bone network architecture

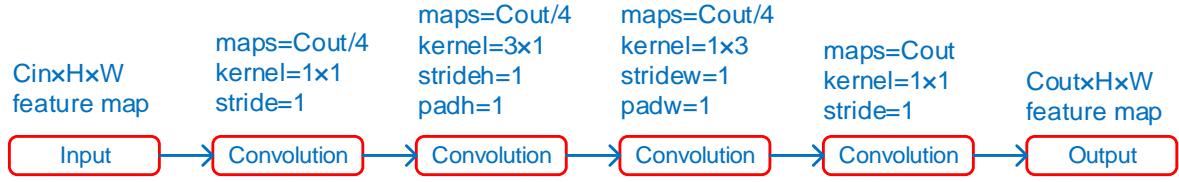


Figure 5. Small basic block

As LPRNet does not implement character recognition after image segmentation, CTC (Connectionist Temporal Classification) loss function without segmentation is used for training [10].

In the decoding process of the inference stage, LPRNet adopts two methods: Greedy Search and Beam Search. Greedy Search takes the maximum class probability of each position and expects to generate a global optimal solution through a locally optimal strategy. Beam Search increases the probability evaluation of Greedy Search. The top n maximum probability combinations are retained in each search, based on which, the final output sequence is obtained by calculating the global maximum probability.

In the post filtering stage, LPRNet uses Beam Search to obtain the top n sequences with the highest probability, and returns the first successful matching template set based on the motor vehicle license plate standard.

2.3. Data set Preparation

The open-source dataset CCPD is the most used open-source license plate dataset with the largest data volume at present, and its main distribution is shown in Figure 6 [11]. Because there are a large number of Anhui type license plates in CCPD, but fewer license plates in other provinces, the license plate data of each province has the phenomenon of class imbalance, which will lead to poor accuracy of Chinese font recognition in other provinces when training license plate recognition network. In order to alleviate this problem, this paper also adopts the method of collecting part of license plates offline, collecting license plate data on the network platform, and generating license plate pictures through mocking the standard style of license plates in the GA36-2018 standard, so as to achieve a relatively balanced data distribution of license plates in various provinces among the yellow, green and blue [12]. Figure 7 , Figure 8 shows the samples.

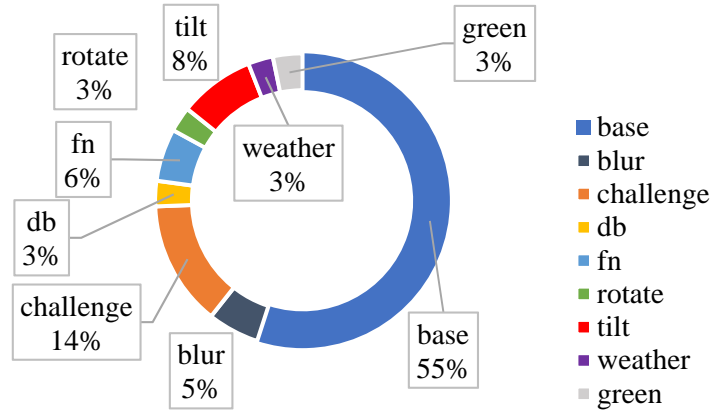


Figure 6. Distribution of CCPD

In the process of generating license plate pieces, first of all, prepare all kinds of license plate bottom frame pictures, yellow and blue license plates using Microsoft Yahei bold font as Chinese font, letters and numbers of them using license plate special font, because green plate is used in different font, collected special Chinese new energy license plate characters, letters and numbers to generate green plates. After generating the standard license plate picture, in order to obtain the real perspective and lighting scene as much as possible, this paper adds random small range perspective transformation, set saturation, brightness, tone; Finally, in order to simulate the stain situation of the license plate, Gaussian noise and stain picture are randomly mixed in the license plate picture. The final resulting license plate effect is shown in Figure 9.

The training dataset of license plate location detection network was randomly selected in proportion from 8 conditions except *ccpd_np* in CCPD, a total of 7440 license plate images were extracted, and another 1488 license plates were selected from *ccpd_base* as the test dataset.

The training dataset of license plate character recognition network has a total of 6971 license plates, including 31 provinces except Hong Kong, Macao and Taiwan, each of which has about 40 yellow, green and blue license plates, and a total of 3720 license plates. A total of 3251 license plates are randomly selected from 8 cases except *ccpd_np* in CCPD according to proportion. In addition, 1469 license plates were selected from *ccpd_base* as the test dataset.



Figure 7. License plate randomly selected from CCPD



Figure 8. License plate collected from offline and online



Figure 9. License plate from mock

2.4. Training Details

By inputting the above training dataset into the network model for training, the characteristic information of the network learning data can complete the correlation mapping between input and output, and the weight parameter can be used to identify the license plate number during detection. The loss curve of the LPRNet model in the training process is shown in **Figure 10**. A total of 100 epochs are trained. With the increase of iteration times, the loss value in the training process first gradually decreases and then becomes stable; in the first 15 epochs, the loss value decreases rapidly; between the 15th epoch and the 50th epoch, the loss value decreases steadily; after the 50th epoch, the loss value decreases slowly, and the performance of the LPRNet model gradually tends to a stable state, and the final loss value stabilizes around 0.023.

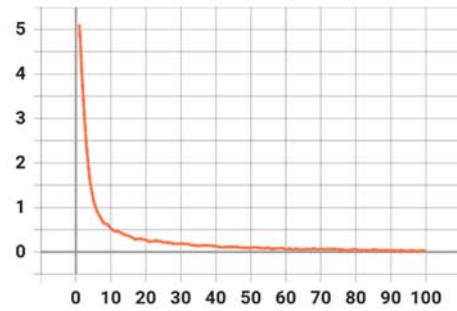


Figure 10. Loss curve

3. Experiments

3.1. License plate recognition process

The trained license plate detection network YOLOv7 was connected with the STN and license plate recognition network LPRNet. The image was input and the detection frame was extracted through the detection network. The output candidate frame was used as the input of the STN. The license plate is corrected by STN, and then the license plate is input into LPRNet. LPRNet first extracts character information from the detected license plate candidate box using the backbone network, and then outputs the final license plate number using CTC, Greedy Search and Beam Search. The overall process and outputs of license plate recognition is shown in **Figure 11**.

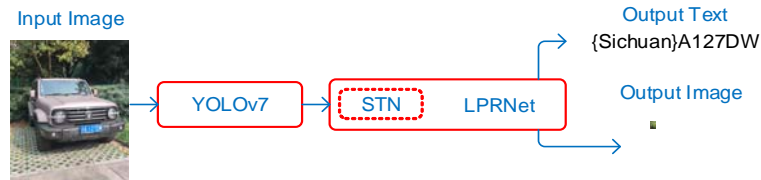


Figure 11. License plate recognition process

3.2. Graphical user interfaces

Qt is a C++ visual development platform, a cross-platform C++ graphical user interface application framework (C++ GUI), capable of providing application developers with the necessary functions to build graphical user interfaces [13]. PyQt is a toolkit for creating Python GUI applications. It is a product of the combination of Qt and Python [14]. It fully integrates the mature Qt user interface library and the excellent features of rapid development of Python language, and is an ideal tool for realizing visual interfaces.

The interface design of this paper is based on PyQt5 implementation. **Figure 12** shows the interface of the visualization program as LPRS (License Plate Recognition System). Click the "Select" button, pop up the system file selection box, and select the directory where the license plate picture is located for detection. After selection, the system loads the first picture of the directory by default, and the pre-marked "Label" is displayed at the bottom left of the interface. Click "Prev" or "Next" button to switch different license plate pictures in this directory, and then click "START" button, the

system will detect the license plate objects in the current picture and identify the license plate number. After the recognition is completed, the license plate number will be printed above the license plate and displayed in "Result".

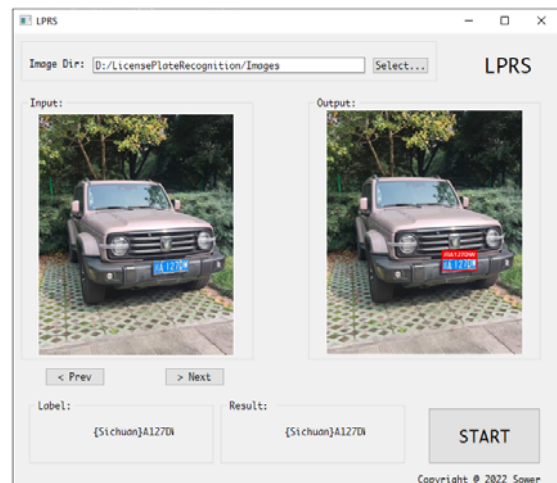


Figure 12. Graphical user interfaces

3.3. Experiment and analysis

License plate location detection belongs to object detection. For object detection models, the evaluation indexes are Recall and Precision, and for license plate recognition, they are the

accuracy of recognition and the overall speed of license plate recognition algorithm [12]. In the test process, the license plate location detection results and analysis, license plate character recognition results and analysis. All neural network training in this paper is carried out based on the laboratory server, and the server hardware and software environment is shown in **Table 1**.

Table 1. LPRS experimental environment

Item	Parameter
OS	Windows 10 Professional
GPU	NVIDIA Ge Force GTX 3070
CPU	Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz
RAM	16GB
Programming Language	Python 3.9
Machine Learning Version	PyTorch 1.13.0 + CUDA11.7

3.3.1. License plate location detection

In the license plate location detection experiment, the detection parameters were set as $img_size=640$, $conf_threshold=0.25$ and $iou_threshold=0.45$. According to the experimental results of the actual scene, the Recall and Precision of the license plate image are as high as 98.33% and 99.55%, respectively. The experimental results show that the license plate detection subsystem can accurately identify the license plate image in the position of sufficient illumination, and shoot the moderate distance, relatively clear license plate image. It shows that the system has high accuracy in complex environment and meets the requirement of high accuracy. The detection result is superior to the traditional algorithm detection method.

In this paper, the the license plate detection subsystem can also effectively recognize license plate images of long-distance, rain, snow and fog weather groups, with Recall of 95.17% and 95.13%, and Precision of 97.77% and 97.79%, respectively. This indicates that the the license plate detection subsystem can also effectively extract the features of license plate images in the face of non-standardized images, complete position regression and realize accurate detection.

3.3.2. License plate character recognition

In the experiment of license plate character recognition, compared with traditional character segmentation methods and other non-end-to-end networks, the processing speed of license plate detection and recognition algorithm based on LPRNet has been greatly improved.

The test results show that the LPRNet has achieved good recognition effect in various complex environment scenes. The recognition accuracy achieving up 96.1%, which indicates that the character recognition system in this paper can be widely used in many scenes. Among them, when the vehicle photo is blurred and the vehicle vision leads to the reduction of the clarity of the license plate, the recognition accuracy of LPRNet is affected the most, which is the main scene of identification errors. The recognition effect is the best in the case of close distance and sufficient light. LPRNet has good recognition effect in the dark light environment, backlight, fuzzy license plate, and the scene of large Angle tilt.

4. Summary

In this paper, the traditional license plate recognition method based on machine vision is easy to be affected by the environment and the recognition effect is not good in complex background environment, etc., the latest YOLOv7 object detection model is adopted for object recognition of vehicle license plate, and then the recognized license plate is cut and input into STN and LPRNet model to realize end-to-end recognition of license plate. Based on the above algorithms, a visual operation program is built on the PyQt5 platform, which can conveniently realize the rapid detection and recognition process of license plate by visual graphical user interface and help realize the intelligent urban security.

The future research direction will improve the license plate recognition algorithm and license plate recognition algorithm, and focus on building an end-to-end independent network from license plate detection to license plate character recognition, so as to improve the accuracy and efficiency of license plate detection and recognition.

Acknowledgment

Project of Education Department of Sichuan Province: Research on Interface Dynamics Modeling of CNC Machine Tool Module (16ZA0070)

References

- [1] Wang Z, Bovik A C, Sheikh H R, et al. Image quality assessment: from error visibility to
- [2] Wang, C. Y. , A. Bochkovskiy , and H. Liao . YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv e-prints (2022).
- [3] Hongwei Wu. License Plate Detection and Recognition System Based on Deep Learning (MA thesis, Dalian University of Technology, China, 2021).p.18.
- [4] Qin Cheng. Research on Fuzzy License Plate Recognition Algorithm Based on Deep Learning (MA thesis, Jilin University, China, 2022).p.15.
- [5] Zherzdev, S. , and A. Gruzdev . LPRNet: License Plate Recognition via Deep Neural Networks. (2018).
- [6] Ting Yu. Research and Implementation of License Plate Recognition Algorithm in Natural Scene based on Deep

- Learning (MA thesis, University of Electronic Science and Technology of China, China, 2022),p.19.
- [7] Yinxun Li. Research and Implementation of License Plate Recognition Technology Based on UAV Perspective (MA thesis, Ningxia University, China, 2022),p.32.
- [8] Jaderberg, M. , et al. Spatial Transformer Networks. MIT Press (2015).
- [9] Srivastava, N. , et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15.1(2014):1929-1958.
- [10] Graves, A. , S Fernández, and F. Gomez . Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. International Conference on Machine Learning ACM, 2006.
- [11] Xu, Z. et al. (2018). Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science(), vol 11217. Springer, Cham. https://doi.org/10.1007/978-3-030-01261-8_16.
- [12] Jiaoyang Tu. License Plate Recognition Algorithm based on Deep Learning in Complex Scenes (MA thesis, Zhejiang University of Technology,China,2020).p.21.
- [13] Information on: <https://doc.qt.io/qt-5.15/reference-overview.html>
- [14] Information on: <https://www.w3schools.cn/pyqt5/index.asp>.