



Research on License Plate Recognition Methods Based on YOLOv5s and LPRNet

Shijian Hu¹, Qinjun Zhao¹(✉), Shuo Li², Tao Shen¹, Xuebin Li¹, Rongyao Jing¹,
and Kehua Du¹

¹ University of Jinan, Jinan 250022, China
cse_zhaoqj@ujn.edu.cn

² Audit Bureau of Postal Savings Bank of China, Jinan 250000, China

Abstract. License plate recognition technology has been applied more and more widely. To meet the speed and accuracy requirements of license plate recognition methods, this paper proposes a license plate recognition method based on YOLOv5s and LPRNet model. First, the YOLOv5s model was used as the detection module, then the detection results were used as the input of the license plate identification module with the LPRNet model as the main part, and finally, the license plate recognition results were output. The practical consequence shows that compared with the other three models for license plate recognition, the recognition method based on YOLOv5s and LPRNet models proposed in this paper has superiorities in the accuracy and speed of license plate identification and the comprehensive identification rate of the license plate is increased to 93%.

Keywords: Deep learning · License plate detection · License plate recognition · YOLOv5s · LPRNet

1 Introduction

Although automatic license plate recognition (LPR) plays a vital role in many practical adhibitions, the current traditional algorithm has great limitations, such as recognition of fixed license plates, limited speed, and fixed background, etc. To reduce the constraints of the working environment, this thesis put forward a new license plate recognition technology, which is composed of a license plate orientation module and a license plate number identification module. The first mentioned of two use the YOLOv5s model to detect license plates from input images, while the latter uses the LPRNet model as the main part, aiming at recognizing numbers and English and Chinese characters in license plates. Model training and data verification are conducted for each module.

2 Related Work

Kanayama, Lee, Matas et al. did extensive research. They used the color attributes and features of the vehicle itself and found that the Sobel operator could produce obvious features at the edge of the image. Through these features can achieve the preliminary license plate recognition effect.

Although these methods play a certain role in identifying deformed and covered license plates, the recognition rate is low. Until the 21st century, with the rapid expansion of machine vision techniques, deep learning, and artificial intelligence technology, the use of neural networks on actual license plate detection has shown better performance and effect. Because vehicle license plates are often affected by stains, illumination, rust, deformation, etc. [1], defects or opaque marks will appear after character segmentation, which seriously affects the accuracy of the feature description. However, Chinese standard vehicle license plates are composed of Chinese characters, English letters, and Arabic numerals, which undoubtedly increases the technical difficulty.

3 YOLOv5s and LPRNet Model Introduction

To improve the tempo and correctness of plate number recognition, this paper chooses the YOLOv5s model in YOLOv5 and LPRNet model to recognize license plates. The method mainly consists of two parts: YOLOv5s is used for image segmentation and license plate position detection. LPRNet recognizes license plates, corrects distortion, and eliminates noise points. YOLOv5s is a single-stage target recognition algorithm that considers performance and accuracy. It increases the recognition rate, guarantees recognition accuracy, and improves the accuracy of the YOLO series for small target recognition.

3.1 YOLOv5s Basic Network Structure

YOLOv5s is made up of Input, Backbone, Neck, and Prediction. The previous algorithm of the YOLO series is to scale the original picture to standard size. It is then fed into the detect network. But in practical application, a lot of photos have a different length-width ratio, so the size of the atramentous edge on both sides will be diverse after zooming and padding. Too much filling will give rise to information redundancy and impact inference speed. So the code in YOLOv5s first modifies the letterbox function, adding adaption to the original image to make the black edge narrower. The mosaic method is then used for data enhancement. Mosaic was proposed in 2019, and it is improved by referring to the data enhancement method of CutMix; instead of selecting two pictures before, four pictures were selected for random cutting, random combination, and random assembly so that a lot of data containing small targets were obtained, which increased the data set and improved the ability to detect small objects [2].

In the Backbone, YOLOv5s mainly adds a Focus construction and CSP construction. The Focus construction is shown in Fig. 1, which is similar to the PassThrough layer of YOLOv3. It converts information from width and length into thoroughfare dimensionality and then separates diverse characteristics by convolution. The Focus structure is used for downsampling (a downsampling in neural networks is mainly used to reduce the number of parameters, reduce dimension and increase local sensitivity). Compared with other algorithms that use a convolution layer and pooling layer whose step size is as small as 2, the Focus layer can effectively reduce the information loss caused by downsampling and reduce the amount of computation [3].

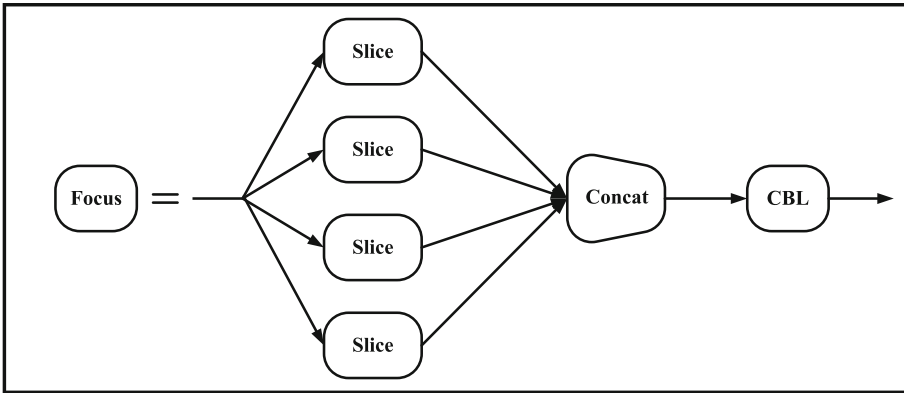


Fig. 1. Structure of Focus

There are two CSP constructions in YOLOv5s. The CSP1_X construction, shown in Fig. 2, is applied to the Backbone, and the CSP2_X construction is applied to the Neck. When the input pictures are 608*608, the variation rule of the characteristic pattern is as follows: 608, 304, 152, 76, 38, 19. After five times of CSP modules, the feature graph with the size of 19*19 is obtained. The CSP structure added to the Backbone network enhances the study ability of CNN, ensuring accuracy while lightweight, reducing computing bottlenecks, and cutting down memory costs [4].

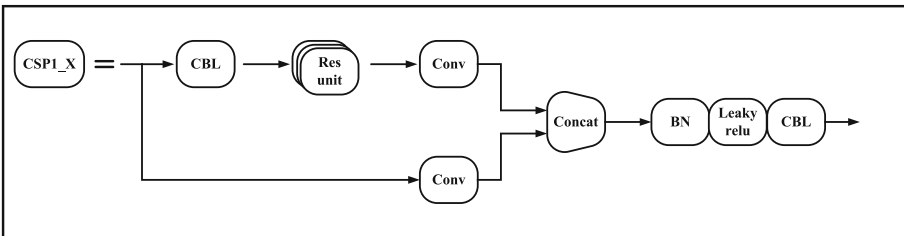


Fig. 2. Structure of CSP1_X

The Neck layer of YOLOv5s employs the construction of FPN + PAN to realize multi-scale feature information fusion. The FPN layer extracts strong semantic features from top to base and fuses the extracted characteristics with those drew by Backbone, while the PAN layer extracts strong positioning characteristics from the base to top and fuses extracted features with those extracted by the FPN layer. Based on the FPN + PAN network construction, characteristics extracted from the Backbone and Neck can be used to improve the characteristic fusion capability of the mesh. YOLOv5s added a CSP2_X structure, as shown in Fig. 3, into the Neck layer to Further improve the capability of network characteristic fusion.

The Prediction layer of YOLOv5s comprises the Bounding box loss function and NMS. IOU is a training parameter. It is the overlap rate between the produced candidate

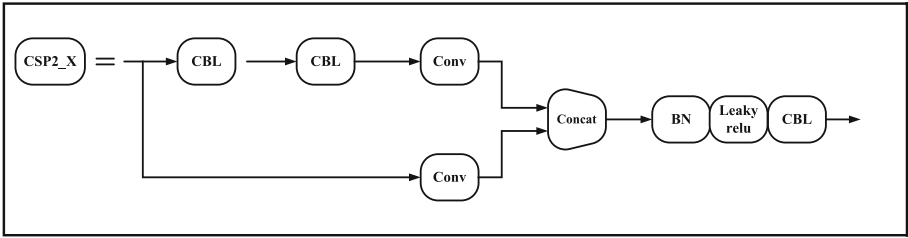


Fig. 3. Structure of CSP2_X

bound and the round truth bound, the value ranges from 0 to 1. YOLOv5s selected $CIOU_{Loss}$ As its Bounding box loss function. Relative to the traditional loss function, the bounding box’s bounding loss function, in essence, optimized the crossover ratio formula:

$$CIOU_{Loss} = 1 - IOU = 1 - \left(IOU - \frac{Distance_2^2}{Distance_C^2} - \frac{v^2}{(1 - IOU) + v} \right) \quad (1)$$

In this formula, v represents the consistency of the length-width ratio, $Distance_2$ is the distance between the prediction box and the center of the standard box, $Distance_C$ is the diagonal distance between the prediction box and the smallest enclosing rectangle of the standard box. Use a weighted NMS (Non-Maximum Suppression) for filtering while generating many target boxes. NMS is generally used to eliminate redundant boxes after model prediction (that is, non-target boxes filtered by the YOLO algorithm after global prediction boxes are generated in the initial state), thus improving speed and accuracy [5].

3.2 License Plate Char Recognition Model Using LPRNet

There are many problems in the course of license plate recognition, such as lighting conditions, weather conditions, image deformation, and license plate tilt Angle. A strong character recognition system should ensure that it can handle recognition tasks in different environments without losing accuracy. LPRNet doesn’t need license plate character segmentation. It is an end-to-end algorithm with excellent adaptation to the environment. The main mesh construction is shown in Table 1. The apiece basic module includes four convolutional layers, one input layer, and one characteristic output layer, and its construction is shown in Table 2. Therefore, establishing the LPRNet algorithm model and applying it to an embedded system can have good performance in detecting relatively complex Chinese license plates. The LPRNet backbone network receives the rawest RGB images as input and calculates the spatial distribution of a great quantity of functions. Wide convolution (1*13 convolution core) replaces the RNN neural network based on LSTM with the context structure of local characters, thus getting rid of the dependence on RNN.

The output of the subnetwork can be viewed as a sequence with probabilities representing the likelihood of the corresponding character, whose length is only equivalent to the breadth of the input picture. Because the decoder output does not correspond to

the length of the objective sequence, the CTC loss function is introduced without the need for segmented end-to-end training [6]. The CTC loss function is diffusely used to solve the input-output order disaccord problem. A raw RGB image is an RGB image with a source network that is used as input to CNN and to extract image features. Use context-dependent 1×13 connections to the kernel instead of LSTM-based RNN. The output of the backbone sub-network can be a sequence representing the related character probabilities, the length of which is related to the width of the input image [7]. Because the network output code isn't equal to the license plate length, this experiment adopts the method of CTC loss for end-to-end training.

Table 1. LPRNet network backbone

Layer Type	Parameters
Input	94*24 pixels RGB image
Convolution	#64, 3*3, stride:1
MaxPooling	#64, 3*3, stride:1
Small basic block	#128, 3*3, stride:1
MaxPooling	#64, 3*3, stride:1
Small basic block	#256, 3*3, stride:1
Small basic block	#256, 3*3, stride:1
MaxPooling	#64, 3*3, stride:2*1
Dropout	0.5 ratio
Convolution	#256, 4*1, stride:1
Dropout	0.5 ratio
Convolution	# class_number, 1*13, stride:1

Table 2. Small basic block structure.

Layer Type	Parameters
Input	$C_{in} * H * W$ feature map
Convolution	# $C_{out}/4$, 1×1 , stride:1
Convolution	# $C_{out} /4$, 3×1 , stride:h:1, padh:1
Convolution	# $C_{out} /4$, 1×3 , stride:h:1, padw:1
Convolution	# C_{out} , 1×1 , stride:1
Output	$C_{out} * H * W$ feature map

3.3 Yolov5s-LPRNet License Plate Recognition Model

Aiming at the complex license plate identification environment, YOLOv5s and LPRNet are combined to design the vehicle license plate recognition system, as shown in Fig. 4. Before the training, the data set is sorted out and reasonably distributed. LPRNet does not need too much training model and only needs a single training for a single license plate type. If the model recognition effect after training is good, the training time can be reduced in the later stage. By connecting the trained license plate detection network YOLOv5s with the character recognition network LPRNet, the image is input, and the detection box is extracted through the detection network; the output candidate box is used as the input of the LPRNet network [8]. The character recognition network LPRNet first extracts the character information from the detected candidate boxes using the backbone network and then uses CTC and cluster to search and output the final license plate number.

Feature extraction is carried out through the LPRNet backbone network, and the license plate sequence is obtained by the convolutional kernel. Due to the difference between LPRNet output and target sequence length, the use of CTC. Using Softmax-Loss to make each character element correspond to each column output, that is, to mark the position of each character in the picture in the training set, this process consumes a lot of time. However, CTC is able to deal with the misalignment of network labels and output [9].

In addition, the detection capability of each node is evaluated by the heuristic function. When the solution space of the graph is very large, the algorithm is often used to reduce the occupied storage space and time resources. When the depth is extended, the nodes with poor quality can be removed, which can reduce space consumption and also effectively save time consumption. LPRNet uses cluster search to obtain the top n sequences with the uppermost feasibility and returns the first successful matching template set, which is based on the motor vehicle license plate standard. Limited by the beam-size argument, the quantity n represents the number of possible words reserved at each step.



Fig. 4. Process of license plate recognition system

4 Experiment and Analysis

The YOLOv5s model and LPRNet model are used to detect the license plate first and then identify the text on the license plate.

First, the overall model of YOLOv5s and LPRNet is constructed to come true end-to-end identification. Then, the overall image is sent to the neural network to come true license plate detection and identification. Finally, license plate positioning accuracy of YOLOv5s is improved by optimizing the loss function and data enhancement in the neural network of YOLOv5s.

In this paper, the CCPD [10] license plate data set is selected, the data is enhanced, and the YOLOv5s model and LPRNet model are retrained, which not only improved the accuracy of experimental results but also prevented the overfitting phenomenon of recognition results. This paper firstly enhanced the data set, then used labelme for data annotation, and finally converted the labelme annotated XML data into the txt file format recognized by YOLOv5 and configured the training parameter table (Table 3).

Table 3. Hyperparameter setting

Parameter name	Parameter setting
img	640*640
Batch size	16
Epochs	300
optimizer	Adam
Initial learning rate	0.10

Where img is the pixel size of the input image, the image input in YOLOv5s must be output in accordance with the specification so that it can enter into the neural network correctly for training; Batch-size is the amount of pictures for a single training; epochs is the number of rounds of training. In the first stage, it took 4 h for training to locate the license plate through YOLOv5. Through data enhancement, the number of data sets is increased, the marking accuracy of data sets is increased, and the model is optimized to raise the overall identification accuracy. In the experiment, 2500 pictures were randomly picked from the CCPD data set as training objects, 80% as the training date, and 20% as the test date. Adam optimizer is selected, and the learning rate decreases with the increase in training times. Taking 100epochs as an example, the change in learning rate is shown in Fig. 5.

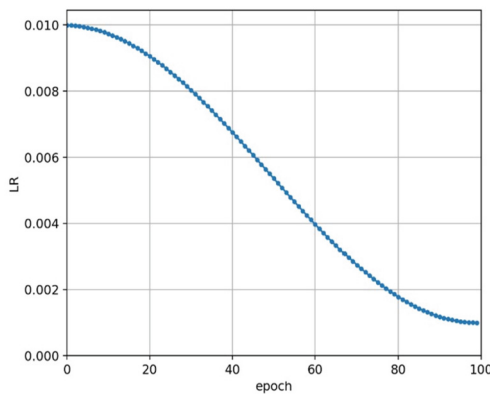


Fig. 5. Learning rate curve

Subsequently, the designed network super parameters are adjusted constantly. The main changes of super parameters are weight attenuation and learning rate increase. The learning rate is an indispensable hyperparameter. Adjusting the learning rate according to training performance can effectively reduce the occurrence of model overfitting.

After 300 epochs of training on the model, the YOLOv5s model has fully entered the convergence state for license plate identification. In the process of model training, accuracy has been steadily improving. The training and test consequence is shown in Fig. 6.



Fig. 6. Test results

It can be seen from the figure above that the recognition effect of the YOLOV5s model is better after 300 epochs, and then the LPRNet model is added, and the recognition result is shown in Fig. 7. The output license plate number is “<Hubei> AC8775”, “<Yunnan> AU7526”, “<Beijing> FV8608”, “<Sichuan> Q79867”, “<Hebei> B911ZB”, “<Fujian> A1035D”, “<Shanxi> K12345”, “<Hubei> AUV027”, “<Liaoning> HK1777”.

The effects of different models on license plate identification are shown in Table 4:

It can be seen that in the CCPD license plate data set, the identification frame rate of the model proposed in this paper on GPU can reach 40 frames, and the recognition accuracy is 3.3%, 9.1%, and 2.1% higher than License_plate, YOLOv4, and YOLOv5, respectively. Through the analysis of the above experimental data, it can be seen that the lightweight YOLOv5s + LPRNet model proposed in this paper greatly improves the speed of license plate recognition on GPU and CPU while improving the real-time of license plate recognition on the premise of maintaining the recognition rate of 93.6%, which has certain practical application value for license plate recognition in future intelligent transportation.



Fig. 7. Identify renderings

Table 4. Each model corresponds to the recognition rate

Model	GPU GTX3060 FPS(figure/s)	CPU I7-12700h FPS(figure/s)	Recognition rate
License_plate	4.41	0.54	90.3
YOLOv4	15.63	1.21	84.5
YOLOv5s	29.12	4.25	91.5
YOLOv5s + LPRNet	40.65	11.65	93.6

5 Conclusion

In the constantly developing modern intelligent transportation system, traffic management, and traffic monitoring, the system can replace some repetitive complex management work; it will save a lot of unnecessary labor and material resources. License plate recognition automation is one of the key technologies for implementing intelligent traffic management. This paper proves that license plate recognition can be realized by using a smaller convolutional neural network. In the complex driving environment, the lightweight LPRNet network selected in this paper has strong environmental adaptability and good anti-interference ability. Combined with the lightweight YOLOv5s network, it can be easily deployed to the hardware on the premise of guaranteeing the recognition rate and accuracy, which can basically meet the requirements for license plate recognition in different environments.

Acknowledgements. This paper is supported by the Shandong Key Technology R&D Program 2019 JZZY021005, Natural Science Foundation of Shandong ZR2020MF067, Natural Science Foundation of Shandong Province ZR2021MF074 and Natural Science Foundation of Shandong Province ZR2022MF296.

References

1. Wang, L., Xu, J., Chen, S.: Design of license plate recognition platform in micro traffic environment. In: 2021 40th Chinese Control Conference (CCC), pp. 6668–6672. IEEE (2021)
2. Zhu, X., Lyu, S., Wang, X., et al.: TPH-YOLOv5: improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2778–2788 (2021)
3. Xiaomeng, L., Jun, F., Peng, C.: Vehicle detection in traffic monitoring scenes based on improved YOLOV5s. In: 2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI), pp. 467–471. IEEE (2022)
4. Xiao, B., Guo, J., He, Z.: Real-time object detection algorithm of autonomous vehicles based on improved YOLOv5s. In: 2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI), pp. 1–6. IEEE (2021)
5. Zhang, X., Fan, H., Zhu, H., et al.: Improvement of YOLOV5 model based on the structure of multiscale domain adaptive network for crowdscape. In: 2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS), pp. 171–175. IEEE (2021)
6. Lamberti, L., Rusci, M., Fariselli, M., et al.: Low-power license plate detection and recognition on a RISC-V multi-core MCU-based vision system. In: 2021 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5. IEEE (2021)
7. Nguyen, D.-L., Putro, M.D., Vo, X.-T., Jo, K.-H.: Triple detector based on feature pyramid network for license plate detection and recognition system in unusual conditions. In: 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE), pp. 1–6. IEEE (2021)
8. Alborzi, Y., Mehraban, T.S., Khoramdel, J., Ardekany, A.N.: Robust real time lightweight automatic license plate recognition system for Iranian license plates. In: 2019 7th International Conference on Robotics and Mechatronics (ICRoM), pp. 352–356. IEEE (2019)
9. Luo, S., Liu, J.: Research on car license plate recognition based on improved YOLOv5m and LPRNet. IEEE Access, vol. 10, pp. 93692–93700 (2022)
10. Xu, Z., Yang, W., Meng, A., et al.: Towards end-to-end license plate detection and recognition: a large dataset and baseline. In: Computer Vision ECCV 2018. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_16