Chinese License Plate Recognition System Design Based on YOLOv4 and CRNN + CTC Algorithm

Le Zhou[1(✉)], Wenji Dai[1], Gang Zhang[1], Hua Lou[2], and Jie Yang[1]

[1] College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
{1319015203,jyang}@njupt.edu.cn

[2] Changzhou College of Information Technology, Changzhou 213164, China

Abstract. License plate recognition (LPR) is widely used in the intelligent transportation systems. Traditional recognition methods have many disadvantages with slow detection speed and low recognition accuracy. In order to solve these problems, this paper proposes an end-to-end LPR method, which is based on YOLOv4 and Convolutional Recurrent Neural Network (CRNN) with Connectionist Temporal Classification (CTC) algorithm, which can effectively improve the detection speed and accuracy. First, based on the excellent classification and detection performance of YOLOv4, it is applied to accurately locate the license plate of the input car image. Then, we use CRNN to recognize the character information imported in the license plate image and add the CTC algorithm to the CRNN network to achieve the alignment of the input and output formats of the character information. Experimental results show that the accuracy rate of license plate recognition detection reaches as high as 97%, and the detection speed is as low as around 30 FPS (Frames Per Second).

Keywords: License plate recognition · YOLOv4 · Convolutional recurrent neural network · Connectionist temporal classification

1 Introduction

In the Intelligent Transportation Systems, the license plate recognition (LPR) system is widely used. LPR analyzes and processes the captured vehicle images under the complex background, so as to obtain the position of the license plate, then automatically recognize the characters on the license plate, and finally output the license plate information automatically. Usually, the license plate is considered as an identity of each vehicle and it is unique. LPR uses this feature of the license plate to identify and count vehicles. In a modern transportation system, the recognition of license plates affects the development of smart transportation. It is also an important factor affecting transportation modernization.

J. Xiong et al. (Eds.): MobiMedia 2021, LNICST 394, pp. 855–865, 2021.
https://doi.org/10.1007/978-3-030-89814-4_63

Traditional LPR methods mainly consist of the following parts: image acquisition, license plate location, character segmentation, and character recognition. First, the position of the license plate is obtained by performing a series of processing on the target image. Second, a certain method is used to divide the characters appearing on the license plate. Third, individual characters are extracted and judged one by one, until the final recognition result is output. Because the license plate has different geometric characteristics, Niu *et al.* [1] binarized the image into black and white and then performed canny edge detection to lock the license plate position. A single character is projected vertically, and the segmented characters are recognized one by one using the template matching method. The experimental results show that the detection performance is well. A. H. Ashtari *et al.* [2] proposed a color-based classification method, which divides it into stable-sized blocks through conversion in the color space. For each block, each small block with the help of a designed filtering process is checked to determine whether it contains a license plate or a certain part of the license plate. Meeras *et al.* [3] proposed the use of three levels of preprocessing local binary pattern classifiers (3L-LBPs) and a large number of AdaBoost cascades to detect license plate regions and improve the speed of license plate detection. Khan *et al.* [4] put forward such a viewpoint that LPR is composed of the following parts: 1) select the brightness channel from the CIE-Lab color space; 2) perform binary segmentation on the selected channel, and then perform image refinement; 3) fuse directional gradient histogram (HOG) and geometric features, and then use a new entropy-based method to select appropriate features; and 4) use support vector machine (SVM) for feature classification, with good results.

In the above-mentioned traditional LPR methods, different angles and positions of the license plates, as well as the accuracy of the license plate character segmentation, have a great influence on the accuracy of the license plate character recognition. The current LPR methods based on the character segmentation methods cannot meet the needs of the practical applications. Therefore, colleagues have proposed an end-to-end LPR algorithm. The advantages of no need for character segmentation, direct input of a complete license plate image at the input, and direct output of recognition results at the output, make end-to-end LPR algorithms highly sought after.

Nowadays, deep learning has been widely used, such as in target detection and classification, safety surveillance [5, 6] and automatic modulation classification [7], as well as in the extraction of abstract and semantic features [8]. Hua *et al.*[9] used deep learning for human emotion recognition (HERO), and the detection accuracy was greatly improved compared with traditional methods. In the wireless signal modulation classification, the automatic modulation classification algorithm based on deep learning has significantly improved the performance and efficiency of the communication system [10], so it is widely used in the field of wireless communication. In the field of intelligent Internet of Things [11], Li *et al.* [8] applied deep learning to remote sensing image classification, which can not only significantly improve the classification accuracy but also enrich the application of deep learning in the field of intelligent Internet of Things. Of course, the license plate detection and recognition system is also an important part of the intelligent Internet of Things. Through studying advanced deep learning algorithms and applying them to the LPR system, not only helps to improve the accuracy of recognition,

but also helps to improve the detection speed, and the operating efficiency of the entire transportation system can also be improved.

Lin et al. [12] proposed the LPR convolutional neural network to improve the character recognition rate of fuzzy images, which does not require character segmentation. Li et al. [13] used VGG to extract low-level CNN features. This method cannot solve the shortcoming of slow VGG network training. Therefore, the method of using a single deep neural network for license plate detection and recognition is not very well. Although the end-to-end recognition framework avoids character segmentation, and it increases the accuracy of LPR system. However, due to the complex combination of Chinese license plate characters and the diversity of shooting angles, the positioning and recognition speed of the license plate is slow and the accuracy is not satisfied.

Based on the deficiencies of the algorithms, this paper proposes a LPR algorithm based on YOLOv4 and CRNN-CTC. The algorithm firstly uses the YOLOv4 network to locate the original image on the license plate detection network and then extracts the image convolution features. With the help of the recurrent neural network (RNN) as the standard model of natural language processing, it can hand text context information very well. Secondly, convolutional features of the images are extracted through the CNN network, then extract the convolution feature sequence of the image. Finally, aiming at the problem that the training characters cannot be aligned, it is solved by introducing the CTC algorithm. In terms of detection speed, the method proposed in this paper is faster. At the same time, the experimental results show that the detection accuracy is also very well.

2 Our Proposed LPR Method

As an application example of character detection and recognition technology, automobile LPR system plays an important role in the construction of smart transportation. The whole system consists of the following parts: license plate location, license plate character recognition, and post-processing recognition and correction, as shown in Fig. 1. This paper puts forward the idea of combining the latest YOLOv4 detection algorithm and the improved CRNN recognition algorithm, which can quickly and accurately detect license plates and recognize license plate characters with excellent performance.

Fig.1. The overall structure of our model

2.1 License Plate Detection Based on Yolov4

For LPR systems, some previous algorithms are based on sliding window search targets that cannot meet the needs. And some improved algorithms on this basis use selective

search to find possible targets such as R-CNN and Faster-RCNN, but the final results are usually determined by CNN or other methods. For the different sizes of photos obtained from different shooting angles and the complexity of the environment, using a sliding window to detect license plates will be very time-consuming and has a high error rate. As shown in Fig. 2, this article uses YOLOv4 [14] to detect license plates. The AP value of the YOLOv4 network developed based on the YOLOv3 network increased by 10%, and at the same time, the corresponding FPS value increased by 12% [15].

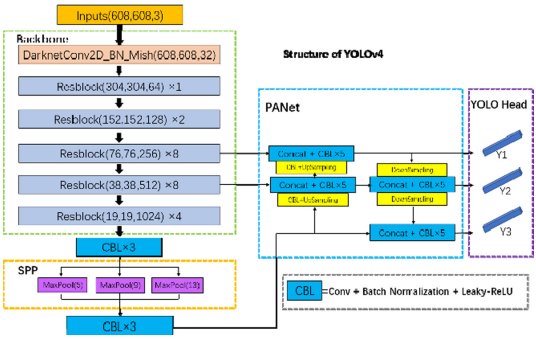

Fig. 1. YOLOv4 network structure.

The model structure of YOLOv4 is shown in Fig. 2. Compared with DarkNet53 of YOLOv3, YOLOv4 uses CSPDarkNet53. And the Neck part is composed of the Spatial Pyramid Pooling (SPPnet) and the Path Aggregation Network (PANet) in the deep convolutional network. The head part is composed of 3 YOLO headers. The CSPDarknet53 network in YOLOv4 is built based on the previous Darknet53 and CSPNet networks, that is, the 5 Resblock bodies in the figure above. The CSPDarknet53 network has many advantages. It can further reduce the amount of calculation and reduce cost. At the same time, the learning ability of the CNN network can be improved, and it also has reliable accuracy. The Darknet53 network with CSP structure is composed of 53 convolutional layers, the sizes of the convolutional layers are 1×1 and 3×3, respectively. In order to better extract the fusion of the target, SPPnet is inserted between the main network and the output layer. At the same time, each convolutional layer is connected to a batch normalization (BN) layer and a Mish activation layer, which can effectively extract target features. It can also further expand the acceptance range of backbone features and play a very important role in separating important context features. PANet is an improved network based on Mask R-CNN. Based on feature fusion, it introduces a bottom-up path augmentation structure. Through bottom-up path enhancement, it reduces the number of convolutional layers that need to pass through the information flow from high-level to low-level. At the same time, the information transmission path is shortened, and the low-level information is transmitted to the high-level, which ultimately makes the positioning information more accurate. And the introduction of adaptive feature pooling makes the extracted ROI features richer. The introduction of fully-connected fusion, which focuses on the overall Context information, and introduces a fully connected branch of the front background two classifications to obtain more accurate segmentation results. YOLOv4

extracts the middle layer in the feature utilization part, the middle and lower layers, and multiple feature layers at the bottom for target detection [16].

In the loss function part, unlike other YOLO models, YOLOv4 uses bounding box regression loss, object classification loss, and object confidence loss. When performing bounding box regression, traditional target detection models (such as YOLO V3), etc. directly set the MSE (mean square error) loss function referring to the center point coordinates of the real box and the prediction box and the width and height information, and then it also uses Intersection-over-Union (IoU) loss instead of MSE, but the performance is not very well. This research uses the latest loss function Complete-IoU (CIoU) [17] of YOLOv4. CIoU considers scale information of the overlap, center distance, and aspect ratio of the frame based on IoU. Such as (1)

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(\boldsymbol{b}, \boldsymbol{b}^{gt})}{c^2} + \alpha\upsilon \tag{1}$$

In this loss function, $\boldsymbol{b}$ represents the center point of the anchor box, while $\boldsymbol{b}^{gt}$ represents the center point of the target box, and $\boldsymbol{\rho}$ represents the Euclidean distance between the two center points. $\boldsymbol{c}$ represents the diagonal distance of the smallest rectangle that can simultaneously cover the anchor box and the target box. Considering the constraint of aspect ratio consistency, CIoU loss adds the aspect ratio constraint $\boldsymbol{\alpha\upsilon}$ to the previous loss. Where $\boldsymbol{\alpha}$ is used as a trade-off parameter:

$$\alpha = \frac{\upsilon}{(1 - IoU) + \upsilon} \tag{2}$$

The parameter $\boldsymbol{\upsilon}$ is used to measure the consistency of the aspect ratio:

$$\upsilon = \frac{4}{\pi^2}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2 \tag{3}$$

where $\boldsymbol{w^{gt}}$ represents the height of the real frame, and $\boldsymbol{h^{gt}}$ represents the width of the real frame, $\boldsymbol{w}$ represents the height of the prediction box, while $\boldsymbol{h}$ represents the width of the prediction box. If the height of the true frame and the predicted frame are similar, then $\boldsymbol{\upsilon} = 0$, the penalty term will not work. So intuitively, in order to better control the height and width of the predicted frame, a penalty item is added to this in order to make it closer to the height and width of the real frame. In this way, CIoU loss considers three important geometric factors for the target frame regression function: overlap area, center point distance, and aspect ratio. Therefore, when solving BBox regression problems, CIoU can achieve better convergence speed and accuracy.

2.2 CRNN Network

The Recurrent Neural Network (RNN) can capture contextual information in sequences. In particular, the two-way neural network can combine historical information and future information to predict the current instance, and it is more effective to use contextual information to perform continuous motion analysis in the time domain than to process each motion separately [18]. The CRNN structure is shown in Fig. 3 consisting of Convolutional Layers, Recurrent Layers, and Transcription Layers.

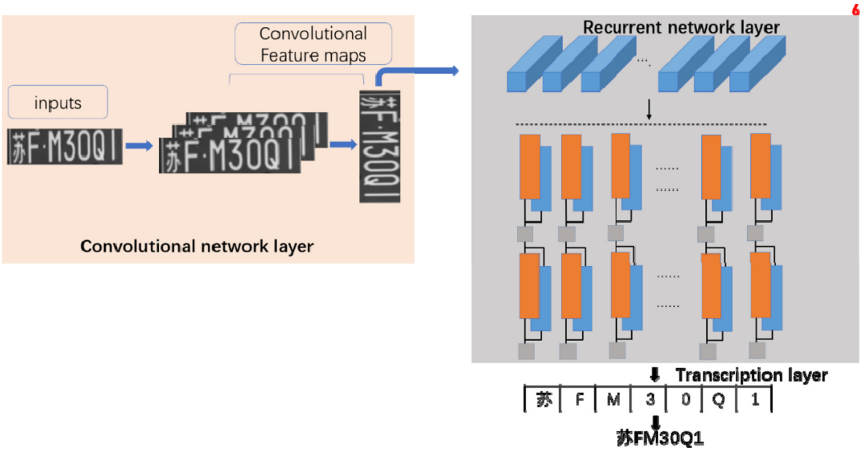

Fig. 2. CRNN Network Structure.

The convolutional layer here is a CNN without the fully connected layer, which is used to extract the Convolutional feature maps of the input image. All images need to be compressed before entering the convolutional layer and then formed into the same size. For the extraction of image feature vectors, the convolutional layer and the maximum pooling layer are mainly used. Then input the obtained feature vector to the loop layer. A deep two-way LSTM network forms a recurrent network layer, which is mainly used to extract features of text sequences. Each two-way memory network contains forward and backward propagation memory networks. The bottom network receives the sequence in the original order, while the top network receives the same input in reverse order. The two networks are not necessarily the same. Importantly, these two-way structures are stacked together, causing their output to be merged into the final prediction. For the predicted value obtained by the loop layer, the transcription layer converts it into a license plate label sequence and finally outputs it. As shown in Fig. 3, after the input image is compressed, it is sent to the convolutional network to extract image features and then converted into a convolution feature matrix. Then it is handed over to a deep two-way LSTM for character sequence feature extraction. Finally, after the RNN output is softmax, it is the character output.

2.3 Connectionist Temporal Classification (CTC)

For Recurrent Layers, if the common Softmax cross-entropy loss is used, each column output needs to correspond to a character element. Then during training, each sample picture needs to mark the position of each character in the picture, and then align it to each column of the Feature map through the CNN receptive field to obtain the label corresponding to the output of the column for training. In actual situations, it is very difficult to mark such alignment samples (in addition to marking characters, but also to mark the position of each character), and the workload is very large. Besides, because the number of characters in each sample is different, the font style is different, and the

font size is different, the output of each column does not necessarily correspond to each character.

To solve this problem, we proposed the use of a connectionist temporal classification (CTC) algorithm. CTC is a loss function. It is based on the concept of a dynamic programming algorithm. It acts on the input and output links of training and only learns the mapping relationship between the input and output links [19, 20, 21]. Only focus on the output sequence we need, without considering the symbol correspondence in the input. It only cares about the convergence of the model to the set of expected sequences and does not care about the region where the symbol is generated. The model can still be trained without knowing the exact location of the symbol corresponding to the ground truth in the input image. Moreover, CTC plays an important role in solving such problems.

3 Experimental Results

3.1 Dataset Generation

In this section, we conducted experiments to verify the effectiveness of the proposed algorithm. We use NVIDIA 1080Ti for this experiment. The environment configurations are Linux Ubuntu 18.04, python 3.6 and pytorch1.3.0. In the experiment, the initial value of the learning rate is 0.001, and it decays exponentially when validation accuracy does not improve in a few previous epochs. The training batch cycle is 100, and each iteration of 100 rounds output a result. The test object is a vehicle license plate with white letters on a blue background of the Chinese mainland. The license plate consists of the Chinese characters representing 31 provinces, the combination of English letters A ~ Z (excluding I and O) and numbers 0–9, a total of 63 characters form a fixed-length 7-digit number license plate. The license plate photos were taken during the day and night, with a total of 20,000 pictures. To ensure that the effect of the test data evaluation is similar to the real scene model, the data set is divided into a training set and a validation set. The training data set is 15,000, the validation set is 1,000, and there are 4,000 test sets.

3.2 Evaluation Criterion and Comparisons

In this section, we compare the results obtained by our proposed method with other state-of-the-art methods through two key factors: the computational complexity of the model and the performance of the model. Due to the complexity of license plate detection, there is no unified established standard for evaluation, so we adopt the evaluation rules of general text detection, that is, we use the accuracy and speed of model detection to measure.

The recognition of license plates in images can be regarded as specific examples of text detection in natural scenes. In our work, the CIoU [17] is used to evaluate the index to evaluate the license plate, referring to formula (1). When the value is 1, the prediction box and manual comment box of the algorithm are completely overlapped, where IoU is represented by formula (4).

$$IoU = \frac{Y}{X} \times 100\% \tag{4}$$

Among them, Y represents the manually marked target box, and X represents the target box predicted by the algorithm.

In order to deepen the understanding of the entire recognition system, we cut out the pictures of license plate detection and divide the experimental results into two parts, license plate detection, and license plate recognition. The following figures (A)-(D) represent the additional measurement results under different angles and light. Picture (A) is the license plate detection result under the standard shooting angle in the daytime; Picture (B) is the license plate taken at a depression angle of 30 degrees during the day; Picture (C) is the license plate taken at the angle of 30 degrees in the daytime; Picture (D) is the license plate taken at a 30 degrees inclination at night.

(A) Standard shooting angle in the daytime.

(B) Overlooking the shooting angle in the daytime.

(C) Oblique shooting angle in the daytime.

(D) Oblique shooting angle at night.

Fig. 3. Partial samples test results.

As can be seen from the above test results, the model proposed in this paper can well detect the target license plate, and the license plate information recognition is relatively accurate. At the same time, the location information of the license plate can be very accurately represented by the preset target box. During the day and night, the inclination angle of 0–30 degrees can achieve better detection results, and the accuracy rate

can reach more than 97%. To further verify the network performance of this experiment, the reference [22] algorithm (YOLO), reference [23] algorithm (Faster R-CNN) and reference [24] algorithm (EasyPR) and the HyperLpr [17] recognition network are compared with the algorithm proposed in this paper. Among them, EasyPR is an open-source Chinese LPR system, developed based on SVM, and the detection accuracy is relatively high. HyperLpr uses the SSD algorithm. It can be seen from Table 1 that compared with EasyPR and Faster R-CNN LPR network, for algorithm detection speed, the method proposed in this article is slightly inferior, but the detection accuracy is greatly improved. Moreover, in the current practical application, 27FPS also meets the number of playback frames of most videos, which fully meets the actual needs of users.

Table 1. Comparison results of different algorithms.

Model	Accuracy (%)	FPS (Frame/second)
YOLO [22]	96.54	–
Faster R-CNN [23]	85.9	102
EasyPR [24]	93	90
HyperLpr [17]	97	15
The proposed algorithm	97	27

4 Conclusion

For the important issue of license plate detection in smart transportation construction, algorithms based on deep learning are always looking for the best detection model. Although various classification and recognition algorithms are emerging in an endless stream, the license plate recognition algorithm based on the YOLOv4 + CRNN-CTC network that we proposed still has certain advantages. Based on the outstanding performance of the YOLOv4 network in image classification and detection, we combine it with the CRNN-CTC network to locate the license plate in the input image and combine the convolutional recurrent neural network (CRNN) and the connection temporal classification (CTC) model to realize the license plate recognition. The recognition process of this method is an end-to-end recognition process and does not need to segment the characters of the license plates, which perfectly avoids the errors caused by the segmentation problem and affects the recognition accuracy. In order to evaluate the performance of this method, the license plates under different light and different inclination angles are used for testing. The experimental results prove the reliability and effectiveness of this method.

References

1. Liu, Y., Yan, J., Xiang, Y.: Research on license plate recognition algorithm based on ABC-Net. In: IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE), pp. 465–469 (2020)
2. Ashtari, A.H., Nordin, M.J., Fathy, M.: An Iranian license plate recognition system based on color features. IEEE Trans. Intell. Transp. Syst. **15**(4), 1690–1705 (2014)
3. Al-Shemarry, M.S., Li, Y., Abdulla, S.: Ensemble of adaboost cascades of 3L-LBPs classifiers for license plates detection with low quality images. Expert Syst. Appl. **92**, 216–235 (2018)
4. Khan, M.A., Sharif, M., Javed, M.Y., Akram, T., Yasmin, M., Saba, T.: License number plate recognition system using entropy-based features selection approach with SVM. IET Image Process. **12**(2), 200–209 (2018)
5. Cao, W., et al.: CNN-based intelligent safety surveillance in green IoT applications. China Commu. **18**(1), 108–119 (2021)
6. Zhao, Y., Yin, Y., Gui, G.: Lightweight deep learning based intelligent edge surveillance techniques. IEEE Trans. Cogn. Commun. Netw. **6**(4), 1146–1154 (2020)
7. Wang, Y., Gui, G., Ohtsuki, T., Adachi, F.: Multi-task learning for generalized automatic modulation classification under non-Gaussian noise with varying SNR conditions. IEEE Trans. Wireless Commu. early access
8. Li, W., et al.: Classification of high-spatial-resolution remote sensing scenes method using transfer learning and deep convolutional neural network. IEEE J. Selected Topics Appl. Earth Observ. Remote Sens. **13**, 1986–1995 (2020)
9. Hua, W., Dai, F., Huang, L., Xiong, J., Gui, G.: HERO: human emotions recognition for realizing intelligent Internet of Things. IEEE Access **7**, 24321–24332 (2019)
10. Wang, Y., et al.: Distributed learning for automatic modulation classification in edge devices. IEEE Wireless Commun. Lett. **9**(12), 2177–2181 (2020)
11. Popoola, S.I., Adebisi, B., Hammoudeh, M., Gui, G., Gacanin, H.: Hybrid deep learning for botnet attack detection in the internet of things networks. IEEE Internet Things J.
12. Lin, C., Lin, Y., Liu, W.: An efficient license plate recognition system using convolution neural networks. In: 2018 IEEE International Conference on Applied System Invention (ICASI), pp. 224–227 (2018)
13. Li, H., Wang, P., Shen, C.: Toward end-to-end car license plate detection and recognition with deep neural networks. IEEE Trans. Intell. Transp. Syst. **20**(3), 1126–1136 (2019)
14. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: optimal speed and accuracy of object detection. arXiv:2004.10934 (2020)
15. Wang, Y., Wang, L., Jiang, Y., Li, T.: Detection of self-build data set based on YOLOv4 network. In: 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE), pp. 640–642 (2020)
16. Li, Y., et al.: A deep learning-based hybrid framework for object detection and recognition in autonomous driving. IEEE Access **8**, 194228–194239 (2020)
17. Qian, Y., et al.: Spot evasion attacks: adversarial examples for license plate recognition systems with convolutional neural networks. Comput. Secur. **95** (2020)
18. Hao, S., Miao, Z., Wang, J., Xu, W., Zhang, Q.: Labanotation generation based on bidirectional gated recurrent units with joint and line features. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 4265–4269 (2019)
19. Liu, H., Jin, S., Zhang, C.: Connectionist temporal classification with maximum entropy regularization. Adv. Neural Inf. Process. Syst. **31**, 831–841 (2018)
20. Feng, X., Yao, H., Zhang, S.: Focal CTC loss for chinese optical character recognition on unbalanced datasets. Complexity 2019 (2019)

21. Miao, Y., Gowayyed, M., Na, X., Ko, T., Metze, F., Waibel, A.: An empirical exploration of CTC acoustic models. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2623–2627 (2016)
22. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788 (2016)
23. Li, Y., Xu, G., Li, W.: FA: A fast method to attack real-time object detection systems. In:2020 IEEE/CIC International Conference on Communications in China (ICCC), pp. 1268–1273 (2020)
24. Xu, M., Du, X., Wang, D.: Super-resolution restoration of single vehicle image based on ESPCN-VISR model. In: IOP Conference Series Materials Science Engineering, vol. 790 (2020)