Chinese License Plate Recognition System Based on Convolutional Neural Network

Hongru Chen [1], Yumin Lin [2], Tianhao Zhao [3,*]

[1] Department of Mechanical and Electronic Engineering, Yanshan University, Qinhuangdao, China

[2.]Department of Information Engineering, Fuzhou University Zhicheng College, Fuzhou, China

[3.]Department of math, Rensselaer Polytechnic Institute, Troy Newyork, United States

* Corresponding Author Email: zhaot4@rpi.edu

Abstract. License Plate Recognition (LPR) technology has been developed for many years, but for the time being, LPR technology still has problems such as not being accurate enough in positioning and too long recognition time. Especially in China, the License Plate (LP) is made up of Chinese characters, alphabets, and numbers, in which the use of Chinese characters profoundly influences the accuracy of LPR. In this paper, a Convolutional Neural Network (CNN) is employed in LPR system, and the LPR system designed in this paper includes three parts: coarse LP positioning, precise LP positioning, and LP character recognition. The LPR system designed in this paper uses the functions in OpenCV to call the laptop camera and input the obtained images into the computer to achieve interframe recognition by setting the timer. Once the image is acquired, the approximate license plate position is determined by the cascade classifier. After this, the image is processed by adaptive binarization, followed by Connected Component Analysis (CCA), linear fitting, and left-right regression to precisely locate the LP. After the exact location of the LP has been determined, it will be fed into a CNN for character recognition. Simulation tests show that the LPR system implemented in this paper has some LPR capability.

Keywords: License Plate Recognition, OpenCV, Convolutional Neural Network.

1. Introduction

As technology for automatic identification of vehicles, Automatic Number Plate Recognition (ANPR) takes part in traffic enforcement [1], vehicle anti-theft [2], and unattended parking lots [3]. The ANPR system will collect license plate images from cameras or other video devices, and give the recognition results after algorithmic processing. The performance of ANPR is disturbed by many factors in practical situations. These factors include lighting, background, picture clarity, license plate style, number of license plates, license plate image distortion caused by shooting, and etc. In previous work, people usually impose restrictions on the working environment of ANPR, such as limiting vehicle speed [4], fixing lighting and limiting the number of vehicles passing at one time. The traditional LPR algorithm usually includes several parts, including LP location, LP Normalization, character segmentation and recognition [5]. The traditional modular ANPR algorithm simplifies the implementation of LPR, but it will be difficult to recognize in some situations. In recent years, computer scholars have used many end-to-end models to address these problems [6, 7]. Related research shows that some end-to-end model can improve the accuracy and speed of recognition. It also has better performance in the case where the traditional algorithm has difficulty in recognition [7]. This research focuses on further improving the recognition performance of ANPR, trying to make ANPR have a higher recognition rate under poor working conditions such as insufficient lighting and picture distortion. Also, we studied dynamic recognition, trying to improve the algorithm's accuracy in dynamic recognition. Besides, we designed the GUI to facilitate the use of the algorithm. In this study, the dataset we use is the Chinese City Parking Dataset, which includes five license plate types. The algorithm mainly includes three parts [8]:

(1) Roughly locate the LP.
(2) Accurately locate the LP.
(3) Character Recognition.

Using the Haar cascade classifier in OpenCV, we were able to roughly find the license plate. For precise locating, we first estimate the upper and lower limits of the license plate using connected component analysis and Random Sample Consensus, then the left and right boundaries using a CNN. For character recognition in the last step, we employ a 17-layer CNN. In the third section we will detail the algorithm's structure and the GUI's design, the fourth section will give the study's findings, and the fifth section will provide a summary of the findings [9].

Automatic license plate recognition is a method that generates vehicle location information by reading license plates on vehicles using optical character recognition on pictures. It may make use of existing installed video surveillance equipment, cameras for law enforcement, or specially made cameras. Such as fixed automatic LPR and mobile automatic LPR, it is a useful and commonly used technology in today's society. But as a result, it has evolved into a tool that needs to fulfill a number of requirements in order to work. Simple image processing methods can be employed in situations with controlled illumination and predictable license plate kinds. To locate license plates in photos, more sophisticated ANPR systems include object detectors like Faster R-CNN, SSDs, and YOLO [10,11].

2. Methodology

2.1. Data

This study used five different types of license plate images from the Chinese City Parking Dataset (CCPD) [7] to implement the LPR system described in this paper. CCPD is an open-source license plate dataset. It contains many LP photos, including fuel and new energy vehicles. The included five types of license plates are CCPD-Base, CCPD-FN, CCPD-DB, CCPD-Tilt, and CCPD-Blur. The categories of the five above-mentioned sections are illustrated in Table.1.

Table 1. The types of data sets

Name	Type description
CCPD-Base	Normal license plate photo
CCPD-FN	Photographing license plates at abnormal distances (too far or too close)
CCPD-DB	Photographs of license plates taken in poor lighting conditions (exposed or dimly lit)
CCPD-Tilt	Photographs of license plates with a tilt angle of 15°-45°
CCPD-Blur	Photographing blurred license plates

2.2. License plate location

Normally, the first link of the LPR system is the license plate positioning. Whether this link can be accurately positioned is related to the ultimate success or failure of the character recognition. The so-called LP positioning is based on the characteristics of the LP, and the image is processed by digital image processing technology in order to obtain a sub-image of the license plate.

Many scholars have been studying this topic for a long time, so there are many proven algorithms for license plate location in the academic community. Such as the cascade classifier, YOLO, SSD, Fast-RCNN, and etc [12-19]. The license plate localization algorithm employed in this paper can be broadly divided into 2 steps. First, the approximate position of the LP is locate by a classifier. This is followed by the use of the CCA, Random Sample Consensus (RANSAC) algorithm, and left-right regression to precisely locate the licence plate position.

For license plate rough localization, we used the Haar cascade classifier in OpenCV. In this process, 700 positive samples and 2300 negative samples were used to train the cascade classifier. Some of the training samples used are illustrated in Figure 1 and Figure 2.



Figure 1. Positive samples



Figure 2. Negative samples

2.3. License plate accurate location

After the rough location of the LP has been completed, the accurate location of the LP is carried out.

In order to reduce the amount of arithmetic involved in subsequent licence plate location and character recognition, the mainstream method is to transform colored images to binary images. After the extensive experimental testing, we decided to use cv.adaptiveThreshold() for the binarization operation. Although cv.threshold() can meet the binarization requirements in most cases, the cv.threshold() function does not perform well when. For example, the license plate image is overexposed and unevenly lit.

As shown in Figure 3, this set of comparison graphs shows the superior ability of the cv.adaptiveThreshold() function in handling overexposed images. The image on the left is an overexposed license plate image, the middle is the image after processing by the cv2.adaptiveThreshold() function, and the right is the image after processing by the cv2.threshold() function.



Figure 3. Comparison chart of binarization operation

In the actual use of the cv2.adaptiveThreshold function, the maximum value was set to 255 after several rounds of experiments. The algorithm used to calculate the threshold is ADAPTIVE_THRESH_GUASSIAN_C. The threshold type is THRESH_BINARY, and the blockSize is set to 27.

After the binarization operation is completed, the connected component analysis is performed. The standard size of Chinese license plate characters is 45 mm in length and 10 mm in height, so this paper uses the aspect ratio of characters as a condition in the CCA. In the CCA, we use the cv2.boundingRect function to get information about the character matrix box and use it as the basis for fitting the upper and lower boundaries.

The RANSAC algorithm has the advantage of being robust and able to estimate the model parameters even in the presence of an insignificant number of outliers. Thus, we use the RANSAC algorithm to fit the up-down boundary. With the RANSAC algorithm, we obtain the up-down boundary of the LP. The next step is to determine the left-right boundary of the LP. In this paper, we use a CNN to perform left-right regression on the image to obtain the left-right edge of the LP. Figure 4 illustrates the structure of the CNN.

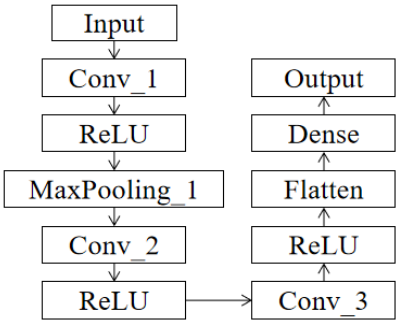


Figure 4. Structure of the left-right regression network

Table 2 illustrates the convolutional and pooling layer parameters in the left-right regression network.

Table 2. Parameters of the left-right regression network

Convolutional layer/Pooling Layer	Kernel Size	Strides	Filters	Pool Size
Conv_1	(3,3)	1	10	
MaxPooling_1				(2,2)
Conv_2	(3,3)	1	16	
Conv_3	(3,3)	1	32	

2.4. Character recognition

In the character recognition process, the recognition method selected and the character recognition success rate have a direct relationship. License plate recognition technology after years of development, there are many mature character recognition methods, such as template matching, SVM, CNN, and so on [20-24].

A 17-layer CNN is used in the LPR system implemented in this paper. We used 7196 character images to train this neural network. In the training process, epochs are set to 30 and batch_size to 6. In this network, we have used the max pooling. The size of each pooling window is 2×2. The structure of this CNN is shown in Figure 5. Table.3 shows the parameters of the convolutional layer of this CNN.

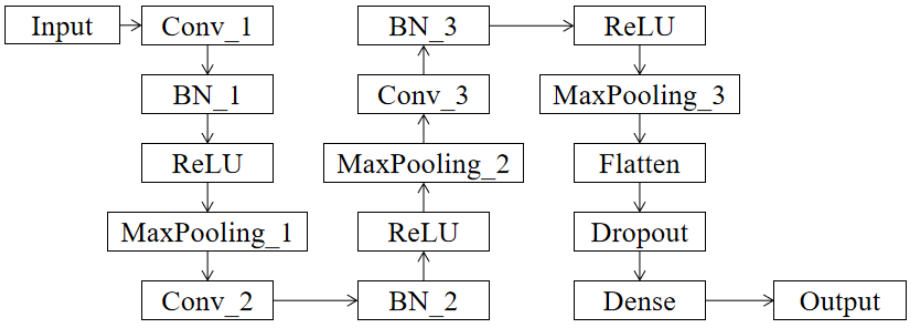


Figure 5. Structure of the character recognition network

Table 3. Structure of the convolutionallayer

Convolutional Layer	Kernel Size	Strides	Filters	Activation Function
Conv2D_1	(3,3)	1	32	ReLU
Conv2D_2	(3,3)	1	64	ReLU
Conv2D_3	(3,3)	1	128	ReLU

2.5. Graphical User Interface (GUI) Design

To facilitate the use of the system by people who do not know the Python language, we have designed a graphical user interface. It can be divided into three sections. The interface is shown in Figure 6. On the left side, there are four buttons, which are the control area. Below that is a text display area which will show the recognized licence plate characters. On the right side is the image or video display area. Figure 7 illustrates the flow chart of the GUI.

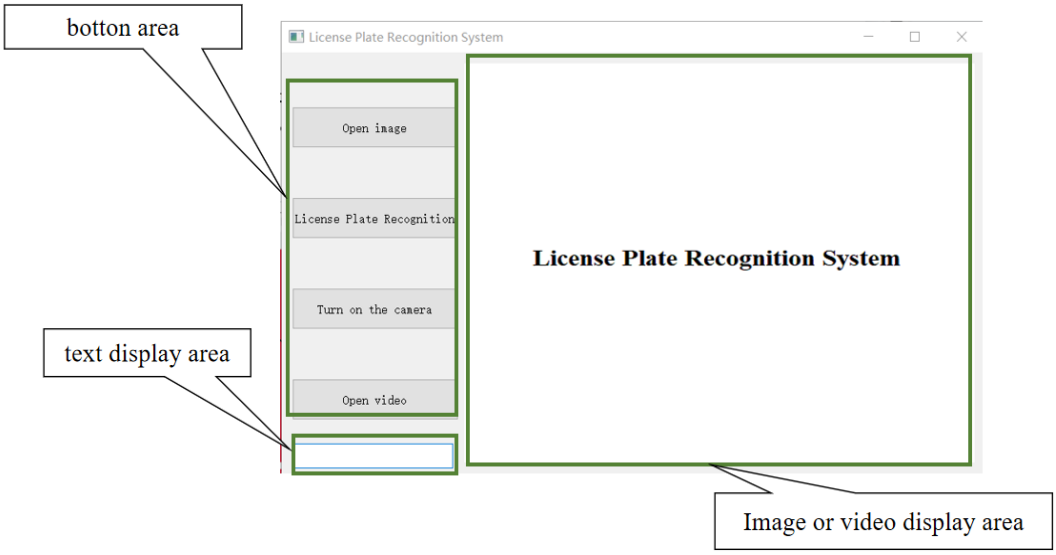


Figure 6. Graphical User Interface

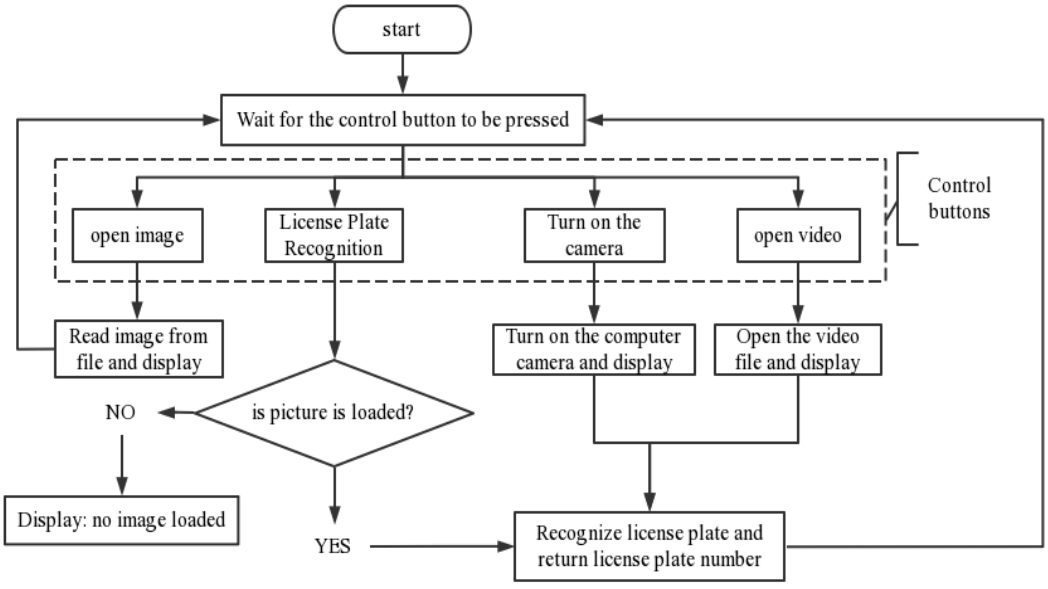


Figure 7. Flowchart of the GUI

Once we start the GUI, it will wait for the control button to be pressed. Once the open image button is pressed, the program is going to open the image from the file and display it in the display area on the right. When the License Plate Recognition button is pressed, it will determine if there is a picture loaded. If so, it will recognize it and display the result. If not, it will indicate that there is no picture loaded. When the open camera button is pressed, it will automatically open the camera and recognize the license plate. When the open video button is pressed, it will open the video file and display it, while recognizing the license plate and returning the result.

3. Results

3.1. Character recognition network training results

The training results are shown in Figure 8. We can see that the loss rate drops rapidly to about 0.4 when epochs are equal to 3, and then steadily decreases to 0.27. In addition, the accuracy rate steadily increases from 0.71 at the beginning to 0.93 through the training process.

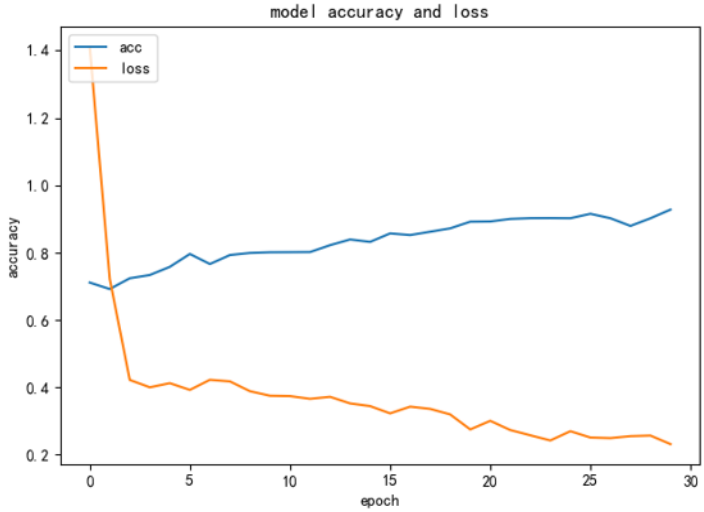


Figure 8. Results of the training

3.2. System test results

To test the LPR system, we conducted static and dynamic tests on the system. In the static test, we selected 1000 images from each of the five data sets: CCPD-Base, CCPD-FN, CCPD-DB, CCPD-Tilt, and CCPD-Blur, making a total of 5000 images as test objects; in the dynamic test, a total of 800 videos of 10 seconds in length were tested. In addition, 200 simulated dynamic tests were conducted using the laptop's own camera. Table.4 illustrates the results.

Table 4. Test results

Name	The number of correct recognitions	Success rate (%)
CCPD-Base	902	90.2
CCPD-FN	701	70.1
CCPD-DB	729	72.9
CCPD-Tilt	688	68.8
CCPD-Blur	449	44.9
Local video text	752	94.0
Simulated dynamic tests	179	89.0

Part of the statically and dynamically tested results are shown in Figure 9.

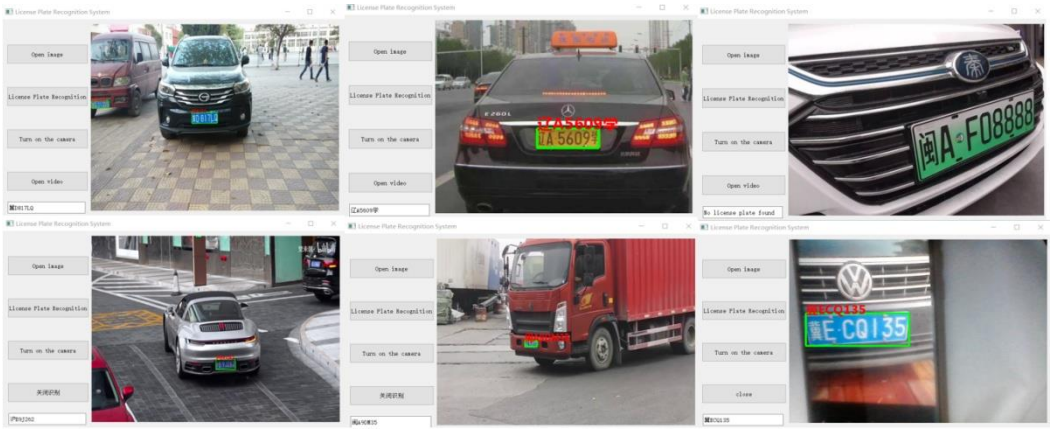


Figure 9. Part of the static and dynamic test results

Through the above analysis, the LPR system implemented in this paper has better performance for both static LP photos and dynamic license plate recognition. Among them, in the static test, for the LP photos taken at a normal angle and having good imaging effect, the recognition rate reaches 90.2%. In addition, the recognition success rate is not good for LP photos taken at a farther or closer distance. In which, for the complete LP shot, the LP shot at a closer distance can be approximately equivalent to the LP shot at a normal distance. However, for LP at a longer distance, there is the problem of low recognition rate. When testing the CCPD-Tilt dataset, for some LP images with too large tilt angle, there is a phenomenon that cannot be recognized. For LP with poor lighting conditions, the recognition rate reached 68.8% due to the adaptive binarization operation. The recognition rate of the LPR system designed in this paper is poor for LP images taken under blurred conditions, which is only 44.9%. Therefore, this part of the recognition algorithm still has scope for improvement. In dynamic recognition, the system has better performance, with a 94% recognition rate for local video and an 89% recognition rate for the simulated LP images.

4. Conclusions

In this paper, we use a CNN in LPR and give a Chinese LPR algorithm. The algorithm is mainly divided into three parts: rough localization, precise localization, and character recognition. We use a variety of algorithms to ensure accurate and efficient recognition. For ease of use, we also designed a GUI. Tests show that the algorithm can perform static or dynamic recognition of license plates. The algorithm has better performance in dynamic recognition. The present system we have created can distinguish different plates accurately, however there are still certain issues that need to be fixed. For instance, the resolution and IR capabilities of the camera are always the primary causes of an incorrect plate. Since the placement, magnification, and pan of the camera always provide different images, this is not the only factor that contributes to a misread number plate. The number plates of the vehicles can be recognized by the ANPR system. As we can see, the camera setup offers the most opportunity for organizations and consumers to improve their ANPR outcomes. It is also important to check the result several times in the engine to get the most accurate result between similar number plates.

References

[1] Davies, P.D., Emmott, N., & Ayland, N.D. (1990). License plate recognition technology for toll violation enforcement.

[2] Lum, C., Hibdon, J., Cave, B., Koper, C.S., & Merola, L.M. (2011). License plate reader (LPR) police patrols in crime hot spots: an experimental evaluation in two adjacent jurisdictions. Journal of Experimental Criminology, 7, 321-345.

[3] Sirithinaphong, T., & Chamnongthai, K. (1999). The recognition of car license plate for automatic parking system. ISSPA '99. Proceedings of the Fifth International Symposium on Signal Processing and its Applications (IEEE Cat. No.99EX359), 1, 455-457 vol.1.

[4] Adorni, G., Bergenti, F., & Cagnoni, S. (1998). VEHICLE LICENSE PLATE RECOGNITION BY MEANS OF CELLULAR AUTOMATA.

[5] Merola, L.M., Lum, C., & Murphy, R.P. (2019). The impact of license plate recognition technology (LPR) on trust in law enforcement: a survey-experiment. Journal of Experimental Criminology, 15, 55-66.

[6] Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A Survey of Deep Learning-Based Object Detection. IEEE Access, 7, 128837-128868.

[7] Xu, Z., Yang, W., Meng, A., Lu, N., Huang, H., Ying, C., & Huang, L. (2018). Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline. ECCV.

[8] PlateRecognizer. 2022. Camera setup for best ANPR: Plate recognizer ALPR. https://platerecognizer.com/camera-setup-for-best-anpr

[9] Wikimedia Foundation. 2022. Automatic number-plate recognition. Wikipedia.https://en.wikipedia.org/wiki/Automatic_number-plate_recognition

[10] PyImageSearch. 2022. OpenCV: Automatic License/Number Plate Recognition (ANPR)with python. https://pyimagesearch.com/2020/09/21/opencv-automatic-license-number-plate-recognition-anpr-with-python/.

[11] Lynch, J., Sheard, N., Gullo, K., Kelley, J., Mullin, J., Galperin, E., Guariglia, M., & Reitman, rainey. 2020. Automated License Plate Readers (alprs). Electronic Frontier Foundation. https://www.eff.org/pages/automated-license-plate-readers-alpr

[12] Masood, S.Z., Shu, G., Dehghan, A., & Ortiz, E.G. (2017). License Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks. ArXiv, abs/1703.07330.

[13] Ibitoye, O.T., Ejidokun, T.O., Dada, O., & Omitola, O.O. (2020). Convolutional Neural Network-Based License Plate Recognition Techniques: A Short Overview. 2020 International Conference on Computational Science and Computational Intelligence (CSCI), 1529-1532.

[14] Silva, S.M., & Jung, C.R. (2020). Real-time license plate detection and recognition using deep convolutional neural networks. J. Vis. Commun. Image Represent., 71, 102773.

[15] Lin, C., Lin, Y., & Liu, W. (2018). An efficient license plate recognition system using convolution neural networks. 2018 IEEE International Conference on Applied System Invention (ICASI), 224-227.

[16] Yousefi, E., Deligani, A.H., Amirbandi, J.J., & Kiskani, M.K. (2019). Real-Time Scale-Invariant License Plate Detection Using Cascade Classifiers. 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 399-402.

[17] Kim, J., & Kim, C. (2021). Implementation of Robust License Plate Recognition System using YOLO and CNN.

[18] Laroca, R., Zanlorensi, L.A., Gonçalves, G.R., Todt, E., Schwartz, W.R., & Menotti, D. (2021). An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO detector. ArXiv, abs/1909.01754.

[19] Islam, T., & Rasel, R.I. (2019). Real-Time Bangla License Plate Recognition System using Faster R-CNN and SSD: A Deep Learning Application. 2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON), 108-111.

[20] Liu, W., & Lin, C. (2017). A hierarchical license plate recognition system using supervised K-means and Support Vector Machine. 2017 International Conference on Applied System Innovation (ICASI), 1622-1625.

[21] Roy, A.C., Hossen, M.K., & Nag, D. (2016). License plate detection and character recognition system for commercial vehicles based on morphological approach and template matching. 2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), 1-6.

[22] Sharma, I., Mathpal, P.C., & Kaushik, A. (2014). Automatic license plate recognition usingoptical character recognition and templatematching on yellow color license plate. International Journal of Innovative Research in Science, Engineering and Technology, 3.

[23] Radzi, S.A., & Hani, M.K. (2011). Character Recognition of License Plate Number Using Convolutional Neural Network. IVIC.

[24] Fang, W., Yi, W., Pang, L., & Hou, S. (2020). A Method of License Plate Location and Character Recognition based on CNN. KSII Transactions on Internet and Information Systems.