Chinese License Plate Recognition Using Machine and Deep Learning Models

2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML) | 978-1-6654-4383-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/PRML52754.2021.9520386

Xiaoyu Zhang
College of Engineering & Science
University of Detroit Mercy
Detroit, USA
zhangxi17@udmercy.edu

Xu Ni
College of Engineering & Science
University of Detroit Mercy
Detroit, USA
nixu1@udmercy.edu

Yanbo Deng
College of Engineering & Science
University of Detroit Mercy
Detroit, USA
dengya@udmercy.edu

Changyu Jiang
College of Engineering & Science
University of Detroit Mercy
Detroit, USA
jiangch@udmercy.edu

Mina Maleki
College of Engineering & Science
University of Detroit Mercy
Detroit, USA
malekimi@udmercy.edu

***Abstract*—The license plate detection and recognition (LPDR) system is one of the practical applications of optical character recognition (OCR) technology in the field of automobile transportation. This paper investigates several state-of-the-art machine and deep learning algorithms for the Chinese license plate recognition based on convolutional neural networks (CNN), long short term memory (LSTM), and k-nearest neighbors (KNN) models. Comparing the performance of these models on the Chinese City Parking Dataset (CCPD) demonstrates that the convolutional recurrent neural network (CRNN) model with an accuracy of 95% is the most accurate and performs better than other models to detect the license plates.**

Keywords—optical character recognition, license plate recognition, machine learning, deep learning, neural networks

I. Introduction

Optical Character Recognition (OCR) is a prominent and active study issue in the realm of image processing and computer vision research proposed by Tausheck in 1929. OCR uses optical and computer technologies to read the text printed or written on paper and convert it into a format that the computer can accept and understand. With the technological advancements, the OCR technology is relatively mature with many different applications and no longer limited to identifying text in books. The automatic detection and recognition of license plates (LP), which is one of the essential modules of intelligent transportation systems, is one of the main practical applications of OCR.

With the improvement of people's living standards, the number of vehicles is gradually increasing. So, the manual management of vehicles is time-consuming and loopholes are easy to appear. As a result, modern advanced equipment and technologies such as OCR have reduced labor-management costs by replacing manual management processes. Furthermore, the automatic license plate recognition system has a wide range of applications, particularly in large cities with complex traffic situations, including road traffic monitoring, traffic accident site investigation, automatic recording of traffic violations, expressway speeding management systems, parking lot management systems, and intelligent community management. Therefore, the automated license plate recognition system has many applications and significant economic values and is worth researching.

AS. Johnson and colleagues developed an automatic recognition system for vehicle license plates using computer vision and image processing technology in 1990 [1]. Since then, various methods, technologies, and algorithms have been developed to detect and recognize LP mainly under well-defined and controlled conditions [1-6] . Some frameworks, for example, require complex hardware to produce high-quality pictures or to record images at a very slow speed from cars [3]. One of the challenges to automatically recognize license plates is under uncontrolled situations that may frequently occur in the real world. The genuinely reliable license plate detection and recognition (LPDR) system should function properly under those conditions. Climatic changes such as foggy weather, deformation, uneven illumination, and blur are some of these uncontrolled situations that make the license plate sensing difficult and inaccurate. Moreover, although many researchers have covered this field of study, it still requires additional research to improve the detection and identification process because of the nature of LP; the numbering system, color, character language, font, and size of LP differ from country to country.

This study surveyed different LPDR models and implemented some of the states of the art machine and deep learning algorithms for license plate recognition. In these methods, after preprocessing the license plate image, including the image segmentation, character segmentation, and character recognition, the license plate number in the input image can be recognized as an output text. Models based on convolutional neural networks (CNN) and its variation (CRNN), long short term memory (LSTM) and k-nearest neighbors (KNN) are some of the implemented models in this study. Data collected from roadside parking in the Chinese City Parking Dataset (CCPD) has been used to evaluate and compare the implemented LPDR models. These models have been compared in terms of detection and recognition accuracies and the amount of their running time.

The paper is organized as follows: Section II discusses the related work, section III describes the research methodology, section IV shows the results of the implemented LPDR models and discussion about the used techniques, and finally, section V concludes a summary of the LPDR models and discusses future work.

II. Related Works

Most papers in the field of automatic LPDR include two stages: LP detection and LP recognition. License plate detection is designed to locate the plate's boundary box in an image. License plate recognition is designed to convert the cut license plate image to text.

You Only Look Once (YOLO) is an algorithm designed to detect different types of objects in a picture or a video [7]. YOLO adopts a one-stage convolutional neural network for the fast prediction of the objects' bounding boxes. YOLO and its more recent variation (YOLOv3) [8] is widely used in many studies to detect the license plate in an image [2, 5, 9]. In [10], a simple model based on OpenCV library along with the python tesseract engine has been used for LP recognition. Google's Tesseract-OCR model is based on LSTM networks [4]. In [11], the author established an end-to-end deep neural network that combines the two-stage of LPDR to one complete algorithm. It has been claimed that identifying the license plate directly from the chaotic picture helps to avoid the accumulation of errors in the middle and speed up the detection and recognition process.

CNN is one of the commonly used deep learning methods used in many studies for both LP detection and recognition [5-6,12]. In [12], a semi-supervised CNN method has been proposed to detect LP anomalies from video frames. In the absence of abnormal data, the license plate recognition system proposed in this study achieves 96% to 99% accuracy. Similarly, in [5], a YOLO object detector has been combined with a CNN model to real-time automatic LP recognition. However, the CNN network named AlexNet used in [6] can recognize the license plate number directly without a detection step.

Convolutional Recurrent Neural Network proposed in [13] combines CNN and LSTM to recognize a string of characters. Later, CRNN was applied to detect Korean cars' license plates from images in [2]. The combination of scene text recognition technique with geometrical image transformation has been employed in this study to recognize number plates for combined neural networks.

The region of interest (ROI) detection method is a more quick way of implementing an automatic number plate recognition system [14]. This model is proposed to avoid character recognition, feature extraction, and comparison of the extracted features with the feature library. Then, the *image Labeller* provided the rectangular ROI label, polyline ROI label, pixel ROI label, and scene label. To detect text and extract the information from an image, image processing techniques were followed. The images collected are together trained in the network model. Similarly, in [15], an ROI algorithm was presented based on mathematical morphology and an edge detection algorithm.

III. Methodology

Most license plate recognition systems have the following processing steps. First, the input images should be preprocessed. Then, after capturing the vehicle in the input image, the license plate boundaries should be detected. After that, a machine or deep learning method should be employed to recognize the characters of the plate. Finally, the recognized characters should be combined into a string to output the entire license plate text.

Preprocessing the input images may include some or all of the following steps: (1) adjusting the image size, which helps avoid some problems with larger resolution images. (2) converting the color image to a grayscale one, (3) image stretching to enhance the contrast between the license plate area and other parts of the image, (4) edge detection to highlight the difference between the license plate boundary and the background.

After preprocessing the images, the license plate should be detected and recognized. This study has implemented four state-of-the-art license plate recognition models based on machine and deep learning algorithms. These methods are briefly described in the following.

A. Tesseract Model

Tesseract, which is an open-source OCR model, can recognize image files in multiple formats and convert them into text [4]. This engine currently supports more than 100 different languages; the version on tesseract 3.01 supports Chinese. Python-tesseract is a wrapper of the Google Tesseract-OCR engine used in this study.

The new version of Tesseract added the deep-learning-based capability with LSTM based OCR engine, which focuses on character patterns and line recognition. The input of Tesseract should be processed images. So, in this study, OpenCV[16] has been employed for image resizing, grayscale processing, edge detecting. Moreover, using a bilateral filter (blur) in OpenCV, unwanted details from the image have been removed. Next, the find Contours function in OpenCV has been employed to find the contour of the license plate in the image. After detecting the location of the license plate, our license plate is cropped from the main image. Finally, the pytesseract package, which is a wrapper of the Google Tesseract-OCR engine, has been used to read characters from the image. A visual schema of this model is shown in Fig. 1.

B. CRNN Model

This model consists of two parts: the license plate detection using the YOLOv3 method and the license plate recognition using convolutional recurrent neural networks. The first part is YOLO which, after training, can quickly recognize a wide variety of objects such as humans, animals, bikes, and even license plates in an image. After detecting the license plate and resizing the detected plate image, the CRNN model [2] is used for LP recognition. As shown in Fig. 2, the implemented CRNN is a combination of CNN and bidirectional LSTM models. The input of this model is a fixed-size license plate (128*64 pixels).

The final output of the CRNN model is a two-dimensional matrix with seven rows corresponding to each LP character and 35 columns corresponding to the value for 26 letters and nine digits characters. The value of each element is between 0 and 1, which is the probability of all possible values corresponding to each character of the license plate.

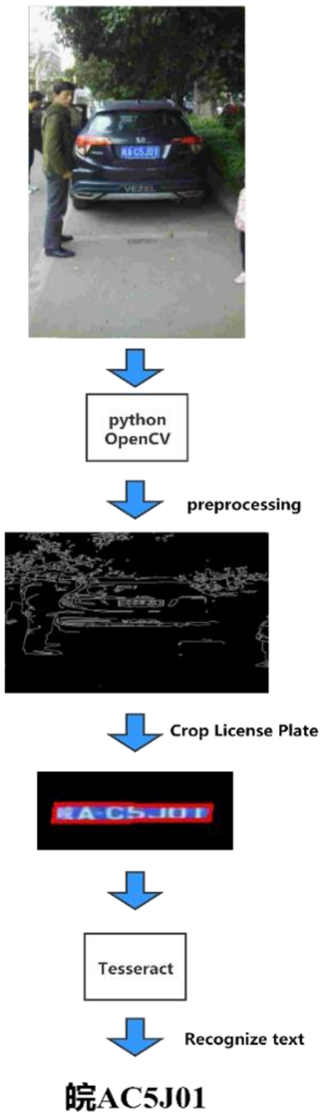

Fig. 1. License plate recognition process using the Tesseract model

C. RPnet Model

The roadside parking net (RPnet) model has two main modules, as shown in Fig. 3. The first detention module is a deep CNN to extract different feature maps from the input LP image. This module has been made up of ten convolutional layers with ReLU and batch normalization, several MaxPooling layers with dropout, and several completely connected layers components. Given an input image, the detection module of RPnet feeds forward the feature map of the last convolutional layer, the box regression layer, to predict the bounding boxes. Assume the bounding box's center point x-coordinate, center point y-coordinate, width, and height are b_x, b_y, b_w, and b_h, respectively. And let W and H denote the width and height of the input image, respectively. The bounding box location c_x, c_y, w, h satisfies in Eq. 1.

$$c_x = \frac{b_x}{W}, c_y = \frac{b_y}{H}, w = \frac{b_w}{W}, h = \frac{b_h}{H}, 0 < c_x, c_y, w, h < 1 \quad (1)$$

The second recognition module is used to extract the region of interest from the input feature map and estimate LPs' number in the input image using multiple classifiers. The extracted ROI from several created feature depictions should be merged and feeds to subsequent classifiers, all referring to the relative location of the bounding box in each feature depiction. After extracting these feature maps, RPnet uses ROI Pooling layers to transform each extracted feature into a feature map with a defined spatial extent of height and width with p channels.

D. ROI-KNN Model

This model, the LP recognition system based on ROI and KNN, is divided into two parts. First, the local license plate area is obtained by pooling the region of interest; the restriction conditions are added to prevent pedestrians or bicycles from being pulled in. Then, the license plate character is obtained by processing images on the license plate area. The processed images have many connected regions. These connected regions can be screened according to the license plate's geometric features to obtain the connected region where the license plate is located. Next, the license plate image is corrected to obtain the license plate binary image. After the image is enlarged again, the connected domain operation is carried out for the binary image. Whether the connected region is a character is judged by the height of the connected domain, and then the upper and lower edges of the character are found and collected. After the segmentation of characters, the neural network is used to recognize the characters. Finally, KNN is used to correct the identified characters.

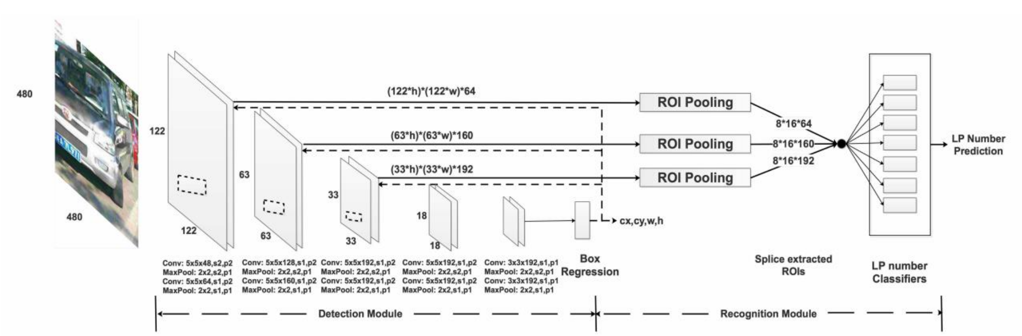

Fig. 2. License plate recognition process using RPnet model [3]

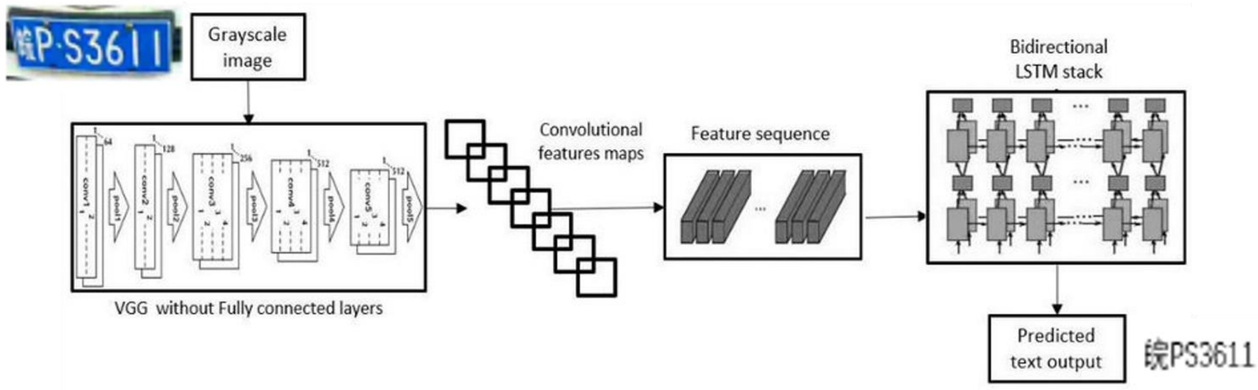

Fig. 3. License plate recognition process using CRNN model [2]

IV. RESULTS AND DISCUSSION

In this section, the performance of four different LPDR models has been evaluated and compared. Accuracy is used as an evaluation metric to compare the LPDR models. In addition, as the detection and recognition speed is also essential, especially for real-time applications, the running time of each model has also been reported.

A. Dataset

CCPD is the dataset used in this study [3]. In this dataset, images were collected from a city parking management company in one provincial capital in China, where car owners own millions of vehicles. The company employs over 800 parking fee collectors, which charge the parking fee on a specific street. CCPD has more than 250,000 pictures in nine different categories. In this study, CCPD-Base images are used to evaluate the LPDR models. Some examples of the CCPD dataset are shown in Fig. 4.

Fig. 4. Dataset example [3]

B. Analysis of Tesseract Model

Based on our experiments, the accuracy of the Tesseract model to recognize the Chinese characters is low. However, its accuracy in recognizing English characters and numbers reached 88.5%. In addition, if the time to load the software library is excluded, it only takes about 244 milliseconds for the model to recognize the license plate number in a picture.

However, this model requires high-quality pictures. Once the picture is blurred or the rotation angle is too large, the model cannot detect the position of the license plate, and the edge detection algorithm of the model failed.

C. Analysis of CRNN Model

As shown in Table I, the accuracy of the CRNN model in the detection stage is 99.8%, while for the recognition stage is 96.6%. So, the detection f the LP in an image using YOLOv3 method is highly accurate. Also, the average time to complete stage 1 is 147 milliseconds, and stage 2 is 426 milliseconds.

TABLE I. PERFORMANCE OF CRNN MODEL

Model	Accuracy(%)	Time (ms)
LP Detection by YOLO	99.8	147
LP Recognition by CRNN	96.6	426

D. Analysis of PRnet Model

By sharing feature mapping between the detection and recognition modules, RPnet is able to achieve a conversion recognition rate of 212 milliseconds per picture and a slightly high recognition accuracy rate of 85.1%.

E. Analysis of ROI-KNN Model

The performance of the ROI-KNN model in different steps is shown in Table II. The accuracy of character recognition is 88%, while the accuracy of character separation reached 95%. The final license plate detection and recognition accuracy using the ROI-KNN model reached 76%. The total amount of time that it takes to recognize an LP from the input picture is 472 milliseconds.

TABLE II. PERFORMANCE OF ROI-KNN MODEL

Experiment	Accuracy(%)	Runtime(ms)
LP Detection	82.07	127
LP Character Separation	**95.01**	117
LP Character Recognition	88.04	228
LPDR	76.0	472

F. Comparison of Recognition Models

The performance of the implemented LPDR methods is demonstrated in Table III. By establishing the model and comparing the accuracy of the model and the running time, we get the following conclusions: (1) the running time of Tesseract model is relatively short, but the accuracy rate is not high, (2) the accuracy rate of the CRNN Model is the highest, but it is also the most complex model among the four models, and the running time is the longest, (3) the PRnet model has the shortest running time, but the accuracy rate is also not high, (4) the recognition rate of the ROI-KNN model is the lowest, and the use time is relatively long, but its recognition rate for a single character is much higher.

TABLE III. MODEL PERFORMANCE

Model	**Accuracy(%)**	**Runtime(ms)**
Tesseract	88.5	244
CRNN	**96.4**	573
PRnet	85.1	**212**
ROI-KNN	76.0	472

V. CONCLUSION

Four state-of-the-art license plate detection and recognition models based on machine and deep learning algorithms have been implemented and compared in this study. It was concluded that among the four models, CRNN is the most accurate, and PRnet is the fastest model for Chinese LPDR.

In future work, the models will be compared on different datasets. In addition, we will also find ways to remove the shortcomings of some models to improve their performance. For example, we will improve the Chinese character recognition ability of the model and improve its image positioning module to improve its ability to recognize lower-quality images.

REFERENCES

[1] M. Fahmy, "Computer Vision Application to Automatic Number Plate Recognition," *IFAC Proceedings Volumes*, vol. 27, no. 12, pp. 169-173, 1994, doi:10.1016/S1474-6670(17)47463-5.

[2] S. Usmankhujaev, S. Lee, S., and J. Kwon, "Korean License Plate Recognition System Using Combined Neural Networks," In *International Symposium on Distributed Computing and Artificial Intelligence*, pp. 10-17. Springer, Cham, 2019.

[3] Z. Xu, W. Yang, A. Meng, N. Lu, H. Huang, C. Ying, and L. Huang, "Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline," In *2018 15th European conference on computer vision (ECCV),* pp. 255-271, Munich, Germany, 2018. doi:10.1007/978-3-030-01261-8_16.

[4] Tesseract OCR. 4.4.1. Google.https://github.com/tesseract-ocr/tesseract

[5] R. Laroca *et al*., "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector," *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1-10, doi: 10.1109/IJCNN.2018.8489629.

[6] S. Lee, K. Son, H. Kim and J. Park, "Car plate recognition based on CNN using embedded system with GPU," *2017 10th International Conference on Human System Interactions (HSI)*, 2017, pp. 239-241, doi: 10.1109/HSI.2017.8005037.

[7] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[8] J. Redmon and A. Farhadi. YOLOv3: An incremental improvement. arXiv preprint, 2018. Available from: http://arxiv.org/abs/1804.02767

[9] R. Laroca, L. Zanlorensi, G. Gonçalves, E. Todt, W. Schwartz, and D. Menotti, "An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO Detector," im *IET Intelligent Transport Systems*, vol. 15. pp. 483-503, 2021.

[10] A. S. Mohammed Shariff, R. Bhatia, R. Kuma and S. Jha, "Vehicle Number Plate Detection Using Python and Open CV," In *International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 525-529, 2021.

[11] H. Li, P. Wang and C. Shen, "Toward End-to-End Car License Plate Detection and Recognition With Deep Neural Networks," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1126-1136, March 2019, doi: 10.1109/TITS.2018.2847291.

[12] B. Kim and J. Heo, "Semi-Supervised Learning Based Anomaly Detection for License Plate OCR in Real Time Video," in *International journal of advanced smart convergence*, vol. 9, no. 1, pp. 113-120, 2020.

[13] B. Shi, X. Bai and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298-2304, 1 Nov. 2017, doi: 10.1109/TPAMI.2016.2646371.

[14] S. Julius Fusic, S. Karthikeyan, H. Ramesh and A. Subbiah, "Vehicle License Plate Detection and Recognition Using Neural Network," *2020 IEEE 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 2020, pp. 1-5, doi: 10.1109/ICCCSP49186.2020.9315206.

[15] Y. Qiu, M. Sun and W. Zhou, "License Plate Extraction Based on Vertical Edge Detection and Mathematical Morphology," *2009 International Conference on Computational Intelligence and Software Engineering*, 2009, pp. 1-5, doi: 10.1109/CISE.2009.5364222.

[16] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, vol. 25, pp. 120-125, 2000.