**Car License Plate Recognition based on OpenCV**

Ng Yun Ning1, Solahuddin Yusuf Fadhlullah1\*

1School of Engineering and Technology, INTI International College Penang, Z-1, Lebuh Bukit Jambul, Bukit Jambul, 11900 Bayan Lepas, Pulau Pinang, Malaysia

\*corresponding author: solah.fadhlullah@newinti.edu.my

|  |  |
| --- | --- |
| **ARTICLE HISTORY** | **ABSTRACT** |
| Received  *201X*  Accepted  *201X*  Available online  *201X* | License plate recognition system autonomously detects the characters on the car plate using camera, be it in static image or video mode. There is a wide field of application available for the implementation of license plate recognition system but most commercial product available lacks the accessibility and flexibility to be utilized in daily lives. This project aims to develop an automated number plate recognition system that offers a wireless interface to local server database which can be easily integrated on mobile devices. The implementation is meant to be lightweight without involving deep machine learning and high processing power. The project investigates on the performance of conventional car plate recognition algorithm under various parameters condition which includes distance, lighting and angle. The car plate recognition system is subjected to 109 sets of practical car plate images and was able to achieve an overall mean accuracy of 91.74%. |
| **Keywords:** *automatic number plate recognition, optical character recognition, Pytesseract, contour approximation, image processing.* |

## 1. INTRODUCTION

### 1.1 Background

Automatic number plate recognition (ANPR) also known as automatic license plate recognition (ALPR) was initially discovered in 1976 by the Police Scientific Development Branch situated in Britain [1]. ANPR system incorporate optical character recognition to identify the alphanumeric character presented on the car plate. Today, as technology advances, ANPR has introduced multiple newer algorithms to aid in the detection and recognition process which aims to increase the accuracy and efficiency of the system. The improvement made on ANPR aims to integrate the technologies to wider application such as car park verification, toll bridges and residential entrance access.

1.1.1 Number Plate Detection

The first step in number plate recognition is on the number plate detection whereby the system uses algorithm to determine the area of interest for data extraction. This process commonly involves the usage of edge detection algorithm to outline the car license plate from the image captured. Edge detection is an algorithm which is used to detect the boundary of image by detecting discontinuation in brightness of the object. Thus, images are generally converted to greyscale before applying edge detection as the brightness can be distinguished easier. There are multiple edge detections algorithms available, each their own specialties and weakness. [2-4]

1.1.2 Number Plate Segmentation

Upon detecting the license plate, the following step is to perform segmentation to section the characters and number for recognition process. To increase the accuracy during segmentation, the quality of the image captured is required to be high. Among the factors that affects the quality includes spatial resolution, image sharpness, contrast, lighting and position and angle of view. [5-6]

Another key element to consider is the noise generated when capturing the image. Despite the factors mentioned above being optimised, noise filtering remains one crucial aspect when performing segmentation to increase character recognition accuracy. One of the most common noise being salt-and-pepper noise which occurs due to sudden disturbance in the image as shown in figure below.

1.1.3 Character Recognition

The last step in automated number plate recognition involves the character recognition process whereby the segmented alphanumeric character is fit to the character recognition algorithm to evaluate on the corresponding license plate number. There are some alphanumeric recognition algorithms that can be used for image processing such as template matching, structural recognition and neural network. [7-9]

## 2. METHODOLOGY AND IMPLEMENTATION

### 2.1 Overall System Design

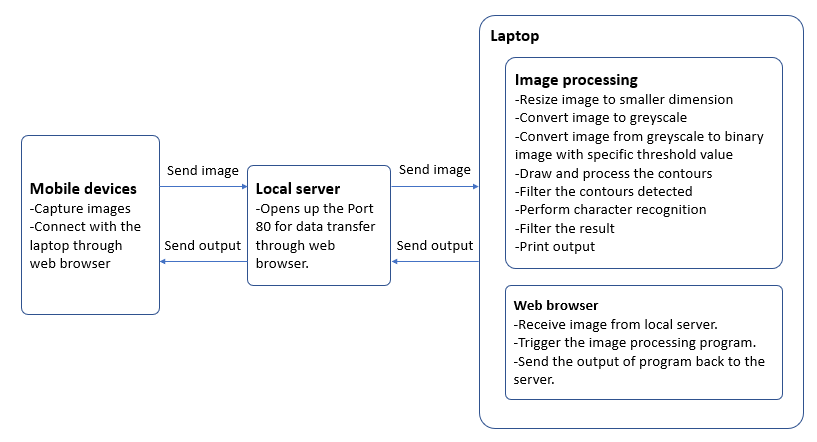


Figure 1: Unified Modelling Language (UML) Diagram of the Car License Plate Recognition System

Figure 1 shows the UML diagram of this project. The laptop serves as the processing unit of this system and is used for implementation of image processing algorithms, serving as the interface between mobile devices and data storing. The camera of mobile device is used to capture the photo of the vehicle number plate and transmit to the laptop through local area network. Before taking a photo, the distance between the vehicle and camera is being controlled to optimize the recognition result. After receiving the image, the image pre-processing will be carried out to reduce noise and improve the efficiency of detection process. During the image pre-processing stage, the input photo will be resized to a constant dimension follow by conversion to grayscale and finally pass through a bilateral filter. The grayscale image will be converted to binary image based on the specified threshold value.

The car plate detection process will be carried out to search for the position of the license plate through contour approximation method. After filtering the contours by using the ratio of the standard car plate, the possible car plates will be identified and cropped out from the original image by referring to the contour coordinate. The brightness and contrast of the cropped image will be adjusted through normalization process. The last stage of the image processing is image recognition, which aims to identify the alphanumeric character on the license plate. A syntactic analysis will be performed on the output of recognition process based on the Malaysia car license plate format. The result which fulfil the predefined condition will be considered as the correct car plate number and displayed on the web browser for the user.

2.2 Software Design

2.2.1 Image Pre-processing

Before an image is feed in the detection algorithm, the input image is required to undergo a series of image processing algorithm that aims to reduce the noise of the image for higher effective detection. The preprocessing steps requirement is tabulated in Table 1.

Table 1: Image Preprocessing Steps and Description

|  |  |
| --- | --- |
| **Steps** | **Description** |
| **Resize** | The resize process aims to standardize the dimension of the input image to allow for greater efficiency during contour approximation. The process also scales the input image to a smaller dimension to reduce the processing time. |
| **Grayscale** | Converting input image to grayscale image that reflects the light intensity of the original image from a value within the range of 0-255. |
| **Bilateral Filtering** | A noise-reducing smoothing filter based on Gaussian distribution that replace the intensity of the pixel with its weighted average to reduces the noises while preserving the edges. This helps to reduce speed up the detection process by reducing the amounts of contours to be identified. |
| **Binary Thresholding** | Creation of binary image from a grayscale image based on its threshold value. This allow for a clear distinction between the light and dark area which allows for proper contour approximation. |

2.2.2 Car Plate Image Detection Algorithm

The car plate detection algorithm used is contour approximation method which is chosen based on car plate geometric size of a rectangular box. Contour approximation method is an algorithm that determine an image object boundary by identifying the continuation of points having the same intensity [10]. The contour approximation is used during the detection stage to identify the dimension of the detected contours. The output from contour approximation can be expressed into one of the two modes which determine how the edges of the approximated contour should be expressed as in Table 2.

Table 2: Contour Approximation Method

|  |  |  |
| --- | --- | --- |
| **Contour Approximation Method** | **NONE** | **SIMPLE** |
| **Description** | All points detected for an approximated contour is shown | Only shows the end points of the approximated contour |
| **Illustration** |  |  |
| **Usage** | Easier visualization of the overall shape | Fewer amount of point to be processed |

The dilation process in morphological operation is used to detect the car plate by searching through the location of the white characters. Dilation operation can wider the white region in an image [11], this property has been implemented on the car plate characters which is white in color. The reason to apply dilation is to make the bright region of the car plate become larger by joining the characters together and make the detection process become easier.

2.2.3 Car Plate Image Recognition Algorithm

The recognition of the car plate images is performed after successful contour detection to reduce the amount of time required. The algorithm developed utilizes the ability of Pytesseract to identify the alphanumerical characters present on the cropped image through bounding box template matching method. This is shown in Figure 2.

The recognition algorithm first determines the number and size of the bounding box for each character detected. Then, Pytesseract performs the recognition based on the dimension as well as the pattern shown registered in it to identify the font and subsequently recognized the characters by matching it to the characters that has the highest likelihood to resemble the character of the bounding box.

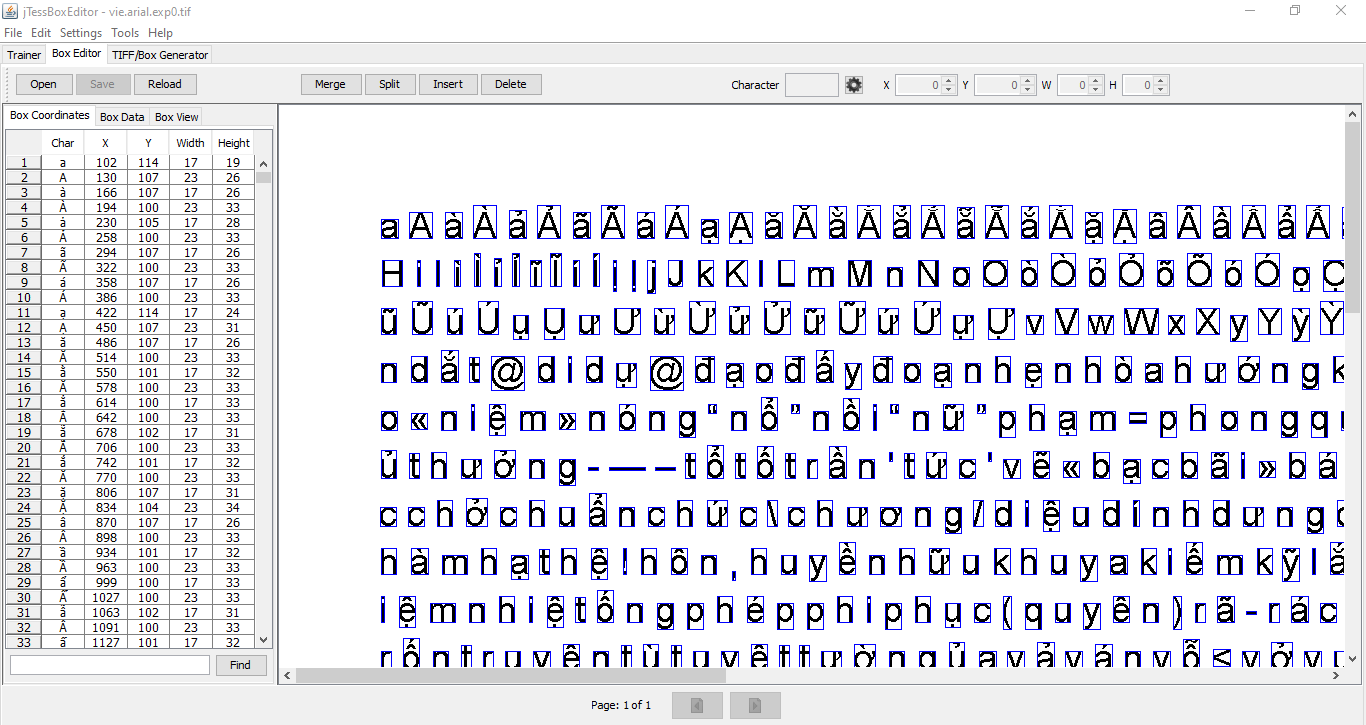


Figure 2: Visualization of Character Bounding Box Dimension Registered in Pytesseract

However, as Pytesseract relies on the dimension and pattern of the font trained, it is highly susceptible to noises from image that affect the recognition performance such as smudges or character disorientation.

2.2.4 Data Interfacing

The data interfacing process has been performed between the processing unit and the mobile device. The camera on the mobile device act as an input which used to capture the image of car plate and send to the laptop to process. After processing, the laptop sends back the result of car plate number to the mobile device and displays on the screen. To connect both devices wirelessly, a local server has been hosted in laptop to enable the mobile device to communicate with the laptop through local area network (LAN). A local area network consists by several computers and peripheral devices that sharing a same communications line or wireless link to a server within a limited distance [12]. A peripheral device which used to send and receive information from computer can be categorized into three main types which is input, output and storage. The two primary methods to allow LAN connections are using Wi-Fi and Ethernet. In this project, the Wi-Fi networking technology that used the radio waves to link the computer and mobile devices to the wireless computer network are chosen to achieve the wireless connection. Thus, all the devices must connect to the same Wi-Fi network in order to communicate with each other.

## 3. results and discussion

### 3.1 Practical Testing

The figure below shows the user interface prototype of the system, which is the main page, the functional page and the output displayed on screen. The picture uploaded will be shown and the car plate number will be prompted as shown in Figure 3.

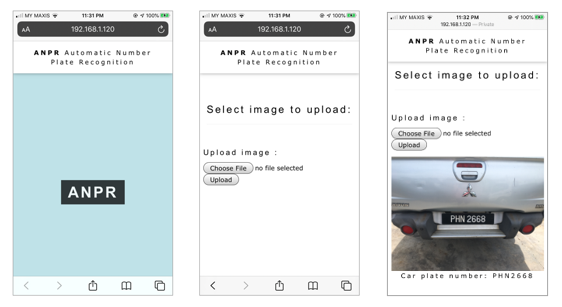


Figure 3: User Interface Prototype of the Car License Plate Recognition System

The table below shows the tabulated result of a number of practical samples used for performance evaluation. The result observed includes the overall process time and accuracy of the recognition system under different settings.

Table 3: System Overall Performance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sample sets** | **Number of images tested** | **Number of image process within 2s** | **Average process time/s** | **Success attempt** | **Accuracy** |
| Indoor Daytime | 15 | 5 | 2.39 | 13 | 86.67% |
| Outdoor Daytime | 46 | 17 | 2.96 | 43 | 93.48% |
| Indoor Night | 9 | 8 | 0.95 | 8 | 88.89% |
| Outdoor Night | 36 | 15 | 2.84 | 33 | 91.67% |
| Special Plate | 3 | 2 | 2.70 | 3 | 100% |
| **Total** | **109** | **47** | **2.70** | **100** | **91.74%** |

From the result shown in Table 3, it can be seen that majority process speed required exceeded the 2 seconds benchmark. The processing speed shown in the table is highly dependent towards the numbers of process undergo to process the input image. The table also indicated that indoor setting images would process faster as compared to the outdoor setting images as outdoor images is subjected to more additional factor as compared to a more controlled environment indoor.

Among the settings shown, car plate images taken when indoor during nighttime has the fastest processing time whereas the accuracy in outdoor setting has a slightly higher accuracy compared to indoor setting which may be contributed by the even lighting from its surrounding compared to indoor setting artificial lighting used. However, this may be due to the short amount of sample sets available thus showing a significant lower process time required compared to the others settings result.

### 3.2 Discussion

Based on the result obtained from each test, it is shown that each parameter tested has impacted the recognition process individually (Figure 4). The distance test allowed for the study on the effect of car plate size towards the recognition rate. The recognition rate can be visualized in the pattern below which can be interpreted as a normal distributed function where the system performance rolls off when the distance range exceeded the upper and lower detection range limit.

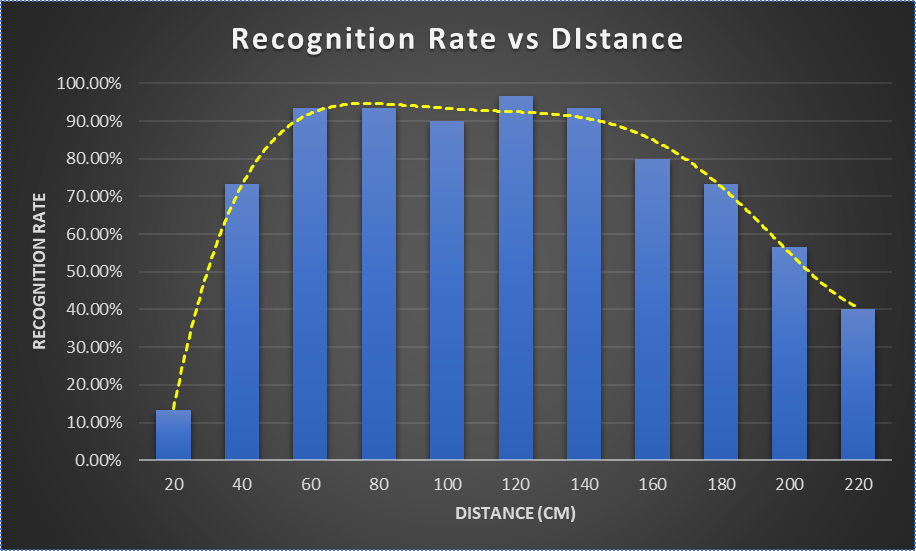


Figure 4: Recognition Rate versus Distance

The second test involves the lighting to determine how lighting would affect the thresholding performance and subsequently affect the recognition performance while comparing it to the distance test carried out. The result shown indicated that the effect of lighting from the image capturing technique is crucial towards the recognition rate even if the surrounding lighting maintains unchanged.

Lastly, the third test carried out tis on the angle test which shown that the tilt angle would also significantly affect the recognition ability as it would distort the alpha numerical character and lead to poor performance when the angle exceeded ±5°. From all three tests carried out, it is concluded that the effect from all three parameters are independent among each other as a defective condition from either one of the parameters would be reflected onto the recognition performance.

## 4. CONCLUSION

Throughout this project, both car plate detection process and character recognition process has been done by using Python programming language and OpenCV image processing library. The results obtained in all experiments has been recorded and analyzed.

In short, the aim of the project is achieved by completed the car license plate recognition system with the accuracy of 91.74%. The effect of distance, lighting and tilt angle of the image towards the accuracy of the system has been explored. Overall, the system can function well in most of the time except those extreme condition such as image captured in very low or high brightness and strongly blurred image. However, there are some future works can be made to improve the design of car license plate recognition system.

## REFERENCES

[1] ANPR International, "*History of ANPR*," Accessed on: June. 20, 2020. [Online]. Available: http://www.anpr-international.com/history-of-anpr/

[2] R. Chandwadkar, S. Dhole, V. Gadewar, D. Raut & S. A. Tiwaskar, "Comparison of Edge Detection Techniques," 2013.

[3] X. Tian, "A Novel Image Edge Detection Algorithm based on Prewitt Operator and Wavelet Transform," *International Journal of Advancements in Computing Technology (IJACT)*, vol. 4, no.19, pp. 73-82, October 2012.

[4] G. T. Shrivakshan & C. Chandrasekar, "A Comparison of various Edge Detection Techniques used in Image Processing," *IJCSI International Journal of Computer Science*, vol. 9, issue. 5, no. 1, pp. 269-275, September 2012.

[5] X. J. He, L. H. Zheng, Q. Wu, W. J. Jia, B. Samali & M. Palaniswami, "Segmentation of Characters on Car License Plates," *2008 IEEE 10th Workshop on Multimedia Signal Processing*, pp. 399-402, 2008.

[6] K. M. Babu & M. V. Raghunadh, "Vehicle Number Plate Detection and Recognition using Bounding Box Method," *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pp. 106-110, 2016.

[7] T. Mahalakshmi, R. Muthaiah & P. Swaminathan, "Review Article: An Overview of Template Matching Technique in Image Processing," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 24, pp. 5469-5473, December 2012.

[8] M. Sabourin & A. Mitiche, "Optical character recognition by a neural network," *Neural Networks*, vol. 5, no. 5, pp. 843-852, 1992.

[9] A. A. Valke & D. G. Lobov, "Character recognition algorithms," *Journal of Physics: Conference Series*, vol. 1210, pp. 1-5, 2018.

[10] OpenCV, "*Contours: Getting Started*," Accessed on: February. 12, 2020. [Online]. Available: https://docs.opencv.org/3.4/d4/d73/tutorial\_py\_contours\_begin.html

[11] OpenCV, "*Eroding and Dilating*," 2019. Accessed on: May. 10, 2020. [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion\_dilatation/erosion\_dilatation.html

[12] M. Rouse, "*local area network (LAN)*," 2019. Accessed on: July. 1, 2020. [Online]. Available: https://searchnetworking.techtarget.com/definition/local-area-network-LAN