Evolution in Electrical and Electronic Engineering Vol. 1 No. 1 (2020) 210 - 218
© Universiti Tun Hussein Onn Malaysia Publisher's Office

EEEE

Homepage: http://penerbit.uthm.edu.my/proceeding/index.php/eeee
e-ISSN : 2756-8458

Vehicle Plate Number Recognition Algorithm for Vehicle Finder System

Xin Yi Liew[1], Siat Ling Jong[1,2*]

[1]Faculty of Electrical and Electronic Engineering,
Univesiti Tun Hussein Onn Malaysia, Parit Raja, 86400, Batu Pahat, MALAYSIA

[2]Wireless and Radio Science Centre (WARAS), Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Parit Raja, 86400, Batu Pahat, MALAYSIA

DOI: https://doi.org/10.30880/eeee.2020.01.01.025
Received 01 August 2020; Accepted 06 September 2020; Available online 30 October 2020

Abstract: Although the smart parking system is a common solution to overcome the issue of parking in the huge shopping mall. However, it encounters a problem on searching their own vehicle in a huge shopping mall which has several levels of parking lots that can accommodate up to hundreds of vehicles. This is due to a visitor has short term memory or easily forget the location of their vehicle parked and lost in the parking lots. Therefore, the visitor needs to spend a lot of time to search for their vehicle based on their short term memory. To this aim, vehicle finder system is proposed to locate the user's vehicle at the car park. The aim of this work is to develop a vehicle finder system which is based on the vehicle plate number. Vehicle finder system has been developed by using Node-RED programming tool and OpenALPR software is used for the plate number recognition. Once the vehicle plate number has been captured, the plate number will be processed and updated to the Google Spreadsheet and then transferred to the bubble.io database for user to check their vehicle number by using mobile applications. Therefore, when the user keys in their vehicle plate number, the details of the location area and route map will be shown in mobile applications. Such system is benefits to the users that has short term memory in locating their vehicle at huge shopping mall. Moreover, it helps users on saving their time in searching their vehicle at huge indoor parking lots.

Keywords: Vehicle Finder System, Plate Number Recognition, OpenALPR, Node-RED

1. Introduction

Finding a vehicle at a huge shopping mall that have several levels of parking lots is such a difficult issue to the person who easily forget the location of their vehicle parked. Although the management of shopping malls use an easy method to let the users or visitors remember the location of their vehicle parked such as numbering, colouring or using picture to distinguish the areas. However, the users still spending their time to locate their vehicle without the help of symbols. Besides that, some of the users remember the symbol or number of the areas that their vehicle parked but they faced the problem on which level do their vehicle placed and which entrance is going in after parked the vehicle. Basically,

**Corresponding author: sljong@uthm.edu.my*

the users will be faced difficulties to search their parked vehicle without using the navigation method [1].

By solving the problem of searching the vehicle in huge indoor parking lots, mobile application is the best choice to overcome this problem which everyone will holding their mobile phone every second. Therefore, mobile application is developed to help the users to locate their vehicle. Most common system to locate the vehicle is global positioning system (GPS) that uses to pinpoint the current location of vehicle and marked on the map to guide the way to take their vehicle [2]. However, this system can not support for indoor parking lots in which the GPS system can not send and receive the signal from satellites due to the attenuation from the roofs, walls and others object. Therefore, the positioning system in indoor is using Wi-Fi as the platform to send and receive the signal.

Regarding to the vehicle finder system, there are several methods proposed before which almost all the proposed methods in each research were based on mobile application. J.Li [3] proposed on car searching system based on smartphone where the users scan the QR code which near to the vehicle parked and the database will generate an optimal car searching path to the users. H.C.Tan [4] presented an intelligent car-searching system based on vision information. This proposed method was focused on calculation of the probabilities of the recognized license plate and the colour of car to sort the retrieval results that the user insert their vehicle condition and it will list out the first 10 results with high similarities to their searching condition. C.L.Liew [5] proposed an indoor parking vehicle locator using QR-code which interact with android. This method is used to provide a navigation process to search the parked vehicles in an indoor parking lots. The application can shows the route map from the current user's location to the location of vehicle parked and also will automatically calculate the steps of user for detecting the current movement.

Other than that, Z.Chen [6] used a reverse car-searching recommendation model to locate the vehicle. The users can scan the QR code after parking which placed on the wall pillars of car parks. The smart-phone will communicate to the central control system to acquire the specific location of the user. When the user wants to take their vehicle, the user can scan any present QR code and then the smart-phone will communicate with the central control system to plan the optimal path which identifying the current location of user and suggest the shortest path to collect their vehicle. Besides, P.C.Lee [7] proposed route guidance of car finding system for an indoor parking lots. This system used a position assisting subsystem and car finding mobile application. A position assisting subsystem acts as an indoor positioning technique by using iBeacon technology while car finding mobile application is used as a media to let users search for their vehicle with the help of navigation strategy.

Furthermore, most of the proposed methods in the research papers are using QR-code as the platform for searching the vehicle parked location. However, the proposed method of this project is based on mobile applications which the users or visitors can search their vehicle by inserting the vehicle plate number in which the users or visitors can save their time on scanning the QR-code after parking the vehicle. The users or visitors also easily forget to scan the QR-code when in a busy moment. The aim of this project is to design a vehicle finder system for indoor parking lots with the use of mobile application as a guidance to pinpoint the location of the vehicle parked. Moreover, this system uses a Raspberry-Pi camera module-v2 to capture and analyse the vehicle plate number and then send it to cloud platform for storage. Therefore, the users can use the mobile application to locate their vehicle by insert the vehicle plate number. The location details as well as map will be shown in the mobile application.

2. Methodology

2.1 System design

The flow chart in Figure 1 shows the implementation of vehicle finder system. Firstly, the motion sensor is in standby mode. When a vehicle is passing through and detected by motion sensor, the Raspberry-Pi 3 will give out signal to capture the vehicle image. Next, the Raspberry-Pi 3 will send a command to OpenALPR software for plate number detection and recognition process. After

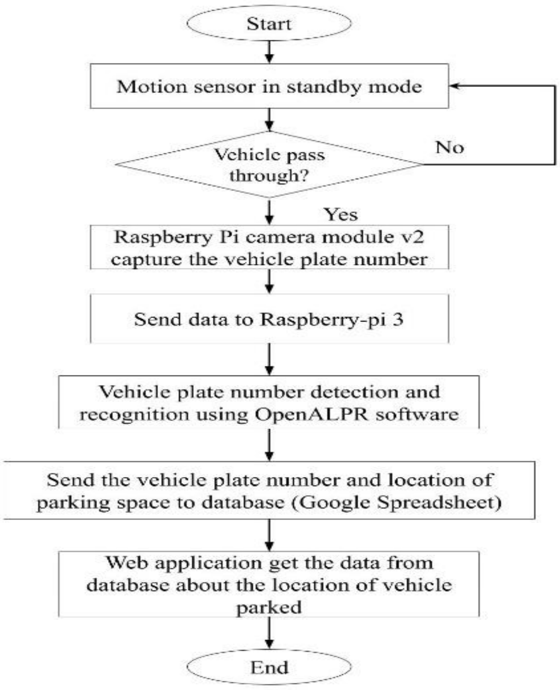


Figure 1: Operation flow chart of vehicle finder system

recognizing the character of vehicle plate, the vehicle plate number and location details will be sent and stored in the database (Google Spreadsheet). Lastly, the users or visitors can search for their vehicle location details through vehicle finder system application and view the location of their vehicle using map. The status of the vehicle finder system will automatically updated to the system if the vehicle is removed from particular parking lot.

2.2 Vehicle finder system using Node-RED

In this work, Node-RED flow is used to implement the vehicle finder system which identifies and sends the license plate number to a database for storage. The Node-RED flow is integrated with the Raspberry-Pi GPIO pins for the input process where the Node-RED flow controls the input devices which are PIR motion sensor and Raspberry-Pi camera module v2 that connected to Raspberry-Pi 3 B+. Figure 2 shows the overall working flow of the system by using Node-RED flow and all the nodes were developed with difference functions.

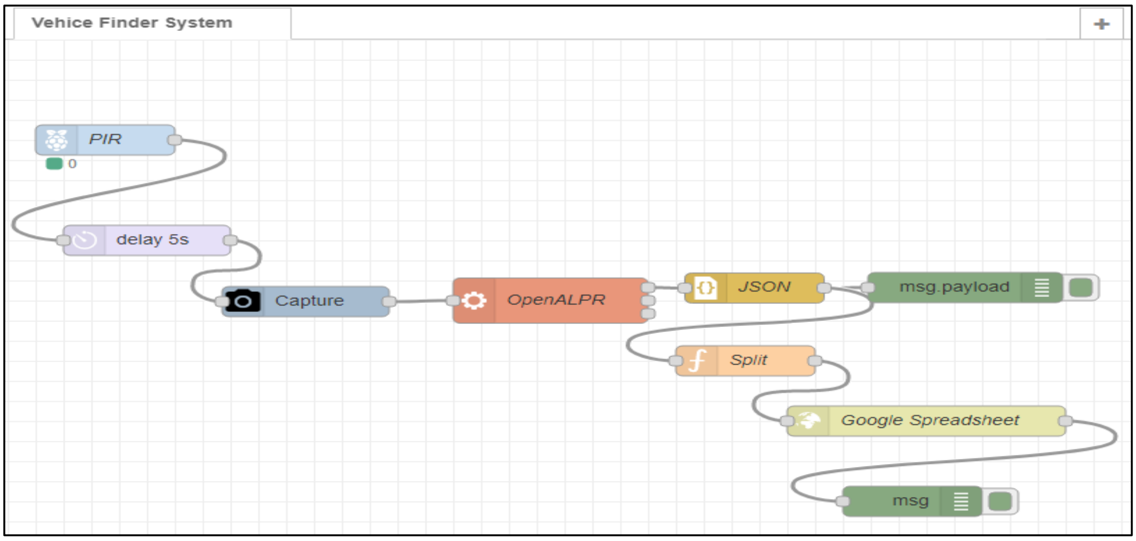


Figure 2: Work flow of vehicle finder system using Node-RED

2.3 Data transfer from micro-controller to server

The server acts as a back-end system which is responsible for receiving data from different current gateways and processing the data for providing required data to the mobile application. The server which is used in this system is Google Spreadsheet. Google Spreadsheet is a fast and easy to use database for collecting and storing a huge amount of data. After the recognition process of plate number, the data of the vehicle plate number will be sent to the server for storage and extracted to mobile application. The data will keep on updating the server and it will show the last location of the vehicle located. Moreover, the data was transferred from Google Spreadsheet to database of mobile application in every 15 minutes by using Integromat software.

2.4 Vehicle finder application

The vehicle finder application is created by using Bubble.io software. This mobile application is a platform for users to have a quick and easy way to locate their vehicle in a huge parking lot in which user only inserts the vehicle plate number and the mobile application will generate the required data such as current location and static route map. The micro-controller will send the data to the server through internet connection from different levels of parking lot and different rows or areas of parking space. The data will be updated to the server and the mobile application will get the latest updated data.

2.5 Plate number recognition using OpenALPR software

OpenALPR is an open source automatic license plate recognition library which is written in C++ language. This library can analyse and identify the vehicle plate number with the input images and video stream and generate the output in text form of license plate characters. Moreover, OpenALPR library uses the concept of the OpenCV and TesseractOCR libraries to run the system. Furthermore, plate number recognition is processed by using OpenALPR software in which the algorithm operates as a pipeline where the input is an image and has various processing occurs in stages and the output of possible plate numbers in the image is extracted. Table 1 [8] shows the pipeline stages of plate number recognition using OpenALPR software.

Table 1: Pipeline stages of OpenALPR software

Pipeline phase	Description
Phase 1: Detection	Finds the potential license plate regions using local binary pattern(LBP) algorithm
Phase 2: Binarization	Converts the plate region image into black and white
Phase 3: Character analysis	Finds the character-sized region in the license plate region by using blobs
Phase 4: Plate edges	Finds the edges of the license plate
Phase 5: Deskew	Remaps the license plate region to a standard size and orientation
Phase 6: Character segmentation	Isolate all the characters in the license plate image and remove the small, disconnected speckles and disqualifying character regions
Phase 7: OCR	Analyses each character image and provides multiple possible plate number
Phase 8: Post processing	Creates a list of plate possibilities based on OCR confidences and determine the best possible plate number combinations

3. Results and Discussion

3.1 Result of vehicle finder system

The hardware of vehicle finder system which shown in Figure 3 is successfully developed by using Raspberry-Pi 3 B+, PIR motion sensor, Raspberry-Pi camera module and interface with Google Spreadsheet, Integromat and Bubble.io of online software.

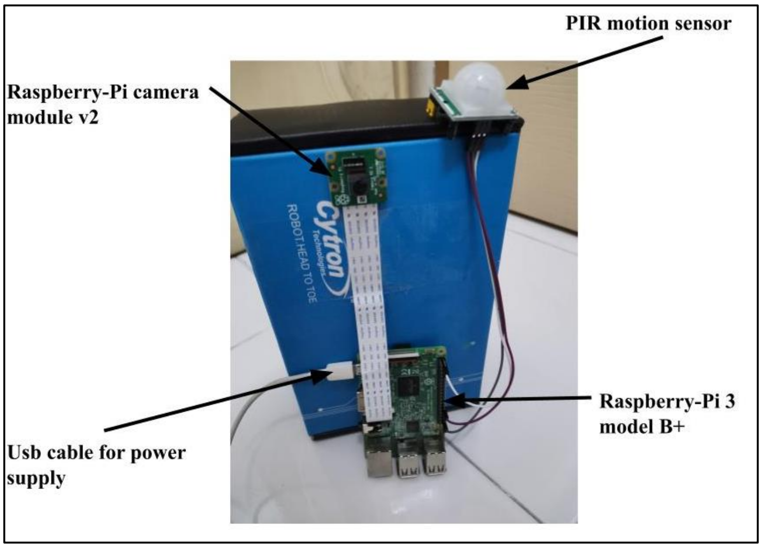


Figure 3: Hardware prototype of vehicle finder system

When the PIR motion sensor detects the motion of an object in the range of 2 to 4 meters, the sensor delays 5 seconds to send a signal to the Raspberry-Pi camera module v2 for capturing vehicle image automatically. Figure 4 shows the result of detection of a PIR motion sensor which shows the digital output signal of 1s at PIN 38.

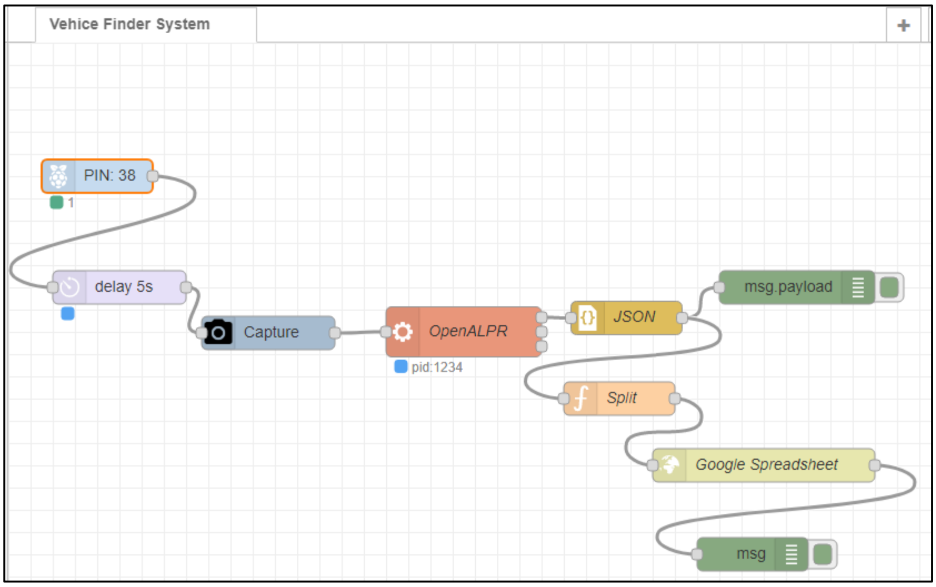


Figure 4: The result of detection of PIR motion sensor at PIN 38

When the camera captures the vehicle image, the system passes to OpenALPR software for the plate number recognition process. After processing the recognition, the result of plate number in the debug dashboard is shown in Figure 5. Figure 6 shows the vehicle plate number which was saved in the Google Spreadsheet. Next, the data in Google Spreadsheet was transferred to bubble.io database by using Integromat in every 15 minutes. Figure 7 shows the plate number was updated to the database of Bubble.io for mobile application user.

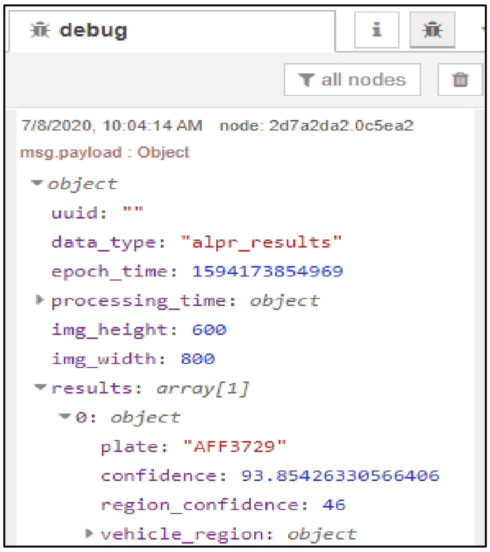


Figure 5: The result of plate number in debug dashboard

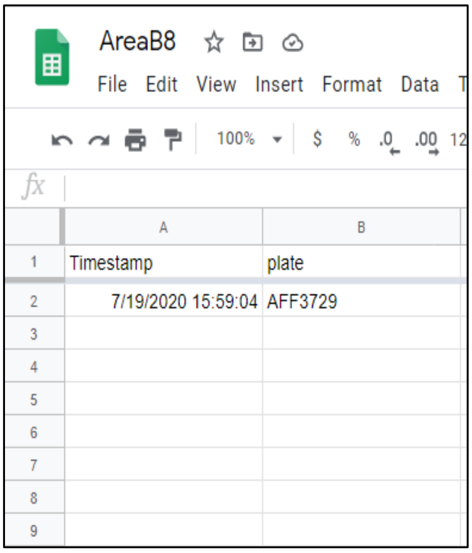


Figure 6: The result of google spreadsheet

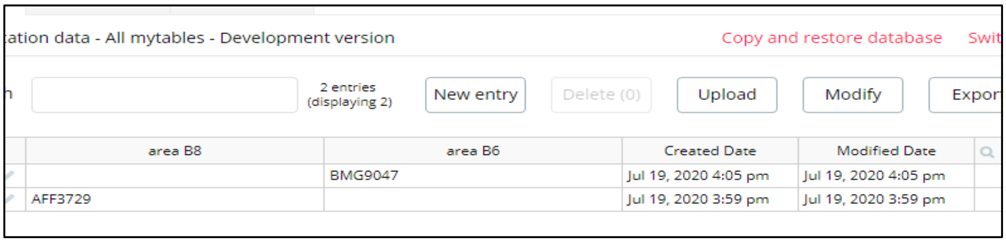


Figure 7: The database of bubble.io software

The last part of the vehicle finder system is the development of the mobile application by using Bubble.io software. When the user keys in their vehicle plate number, the plate number, location details, time stamp and static map were displayed in the mobile application that is shown in Figure 8. If the user keys wrongly or the plate number was not in the database, the text of "no car found" was displayed which is shown in Figure 9.

Figure 8: The display on mobile application

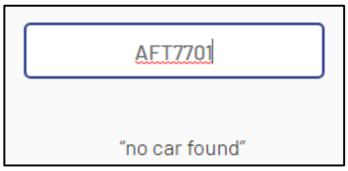
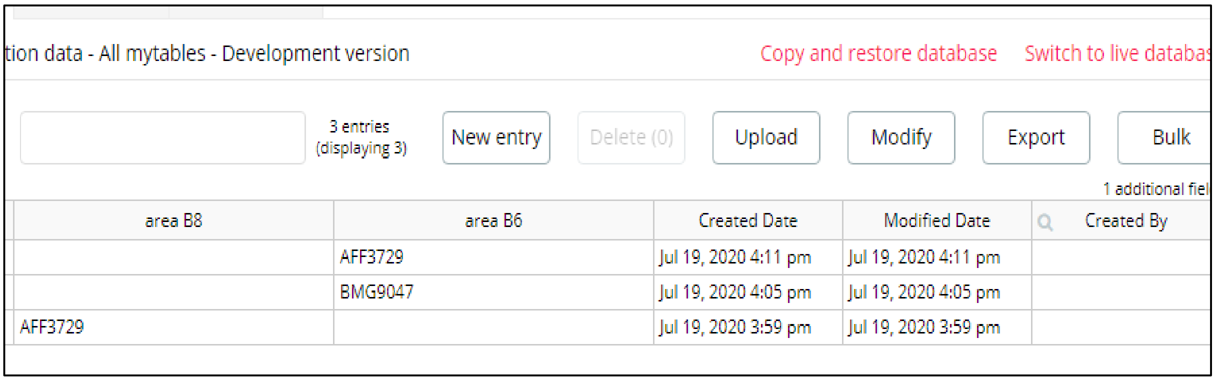


Figure 9: The display on mobile application when vehicle plate number was not found

On the other hands, when parking space of area B8 are fully parked, the vehicle will be moved to another parking area such as area B6 to search for empty parking space in which the camera at area B6 captured the vehicle of "AFF3729" and updated to the Bubble.io database which is shown in Figure 10. Therefore, the mobile application will display the latest location detail when the user key in their vehicle plate number which shown in Figure 11.



area B8	area B6	Created Date	Modified Date	Created By
	AFF3729	Jul 19, 2020 4:11 pm	Jul 19, 2020 4:11 pm	
	BMG9047	Jul 19, 2020 4:05 pm	Jul 19, 2020 4:05 pm	
AFF3729		Jul 19, 2020 3:59 pm	Jul 19, 2020 3:59 pm	

Figure 10: The database of bubble.io software

Figure 11: The display on mobile application

3.2 Performance test of plate number recognition using OpenALPR software

- The distance between camera and vehicle

Table 2 shows the results of the plate number recognition process by using OpenALPR software on various distances. The distances under test are 1 meter, 2 meters, 3 meters, 4 meters and 5 meters and the camera was placed at horizontal sight with the vehicle. As shown in Table 2, the maximum distance of the captured vehicle image which can accurately extract the plate number of "AFF3729" is 4 meters while the minimum distance is 1 meter. At a distance of 5 meters, the camera captured the vehicle image and extracted a plate number with the result of "AF3729" which is incorrect. This is due to the further the distance of the vehicle from the camera, the sharpness of the captured image will be greatly reduced. Thus, the accuracy of the recognition will be decreased. Therefore, the sharpness of the captured image depends on the quality of the camera.

Furthermore, the size of the license plate "BMG9047" is smaller compare with the standard size of license plate. In this analysis, the maximum distance of captured vehicle image from the camera to vehicle is 3 meters whereas the minimum distance is 1 meter which is shown in Table 3. At the distance of 4 meters, the license plate of "BMG9047" was extracted incorrectly in which the character 'G' was extracted to number of 6. This due to the character of 'G' is look similar to character number of '6' which cause by the quality or captured image.

Figure 12: License plate of "AFF3729" and "BMG9047"

Table 2: The result of plate number recognition based on various distances

Distance	Actual number	Recognized number	Remark
1 meter	AFF3729	AFF3729	Correct
2 meters	AFF3729	AFF3729	Correct
3 meters	AFF3729	AFF3729	Correct
4 meters	AFF3729	AFF3729	Correct
5 meters	AFF3729	AF3729	Incorrect

Table 3: The result based on distance and license plate "BMG9047"

Distance	Actual number	Recognized number	Remark
1 meter	BMG9047	BMG9047	Correct
2 meters	BMG9047	BMG9047	Correct
3 meters	BMG9047	BMG9047	Correct
4 meters	BMG9047	BM69047	Incorrect

- Character types of license plate

By using OpenALPR software, three types of license plates are being taken to analyse the performance of the recognition software. As shown in Table 4, the result of the recognition with different types of license plate is analysed. The license plate of "BMG9047" was successfully extracted while the license plate of "ALA7215"and "JAG3962" were unable to be recognized. This is due to the types of license plates that vary from the standard license plate and the recognition software was unable to search and compare the samples of the character.

The license plate of "JAG3962" is in Italic form and the gap between each of the character was smaller whereas the license plate of "ALA7215" is bold and the character 'A' is different from the standard character of 'A'. Moreover, the characters of "BMG9047" license plate are almost similar to the standard license plate. However, the gap between the character and the size of character is smaller compared to the standard license plate. Therefore, the different types of the license plates would be difficult to extract correctly when the recognition system did not have those samples of license plate character.

- Accuracy rate of plate number recognition

Table 5 shows the result of accuracy rate of the plate number recognition with four different types of vehicle plate number. The analysis of distances between camera and vehicle is fixed at 4 meters and the total of 10 test samples were analysed. As shown in Table 5, the accuracy rate of license plate "AFF3729" being extracted was 100% and recognized correctly. However, the accuracy of the license

Table 4: The result based on different types of license plate

License plate	Actual number	Recognized number	Remark
	BMG9047	BMG9047	Correct
	ALA7215	Unable to read	-
	JAG3962	Unable to read	-

plate "BMG9047" being extracted was 70% due to the character types of license plate "BMG9047" was almost the same as the standard license plate, thus the plate number can be successfully extracted. However, low quality of captured image could be another reason that the plate number unable to be extracted correctly. Moreover, the size of the characters and the gap distances between each character were smaller than the standard license plate could affect the recognition result, therefore it only shows 30% of the accuracy.

Besides that, the license plates of "ALA7215" and "JAG3962" were unable to be extracted and recognized correctly which is 0% of accuracy rate. Therefore, the plate number recognition was failed from the fancy types of plate numbers and the far distances between vehicle and camera which is more than 4 meters. This is due to lack of sample types of the character in the database of OpenALPR software for the recognition process. Furthermore, the presence of blurry and noisy captured image would affect the accuracy of the recognition result when the distance between vehicle and camera is far. However, the results show a high recognition result for the standard license plate which is 100% whereas 0% for fancy types of plate number and 30% for the blurry and noisy containing captured images.

Table 5: The accuracy rate of plate number recognition

Plate number	Number of test result		Accuracy (%)	
	Extracted	Correct recognized	Extracted	Correct recognized
AFF3729	10	10	100	100
BMG9047	7	3	70	30
ALA7215	0	0	0	0
JAG3962	0	0	0	0

4. Conclusion

In conclusion, a vehicle finder system is developed to overcome the problem of searching the vehicle in a huge indoor parking lot. This system is well-designed by using the Node-RED tool to wire all the hardware devices, API and online software together to deploy it in the Raspberry-Pi 3 model B+ as a standalone hardware system. Moreover, the PIR motion sensor able to detect the motion and trigger the camera to capture the vehicle image. The OpenALPR software able to recognize the license plate and save into the Google Spreadsheet for storage. Next, the Integromat software would automatically update the data in every 15 minutes to the database of Bubble.io. For the mobile application, the users able to search for their vehicle by key in the vehicle plate number. At the same time, the location details and route map of the vehicle placed is displayed. This system is beneficial to users who park their car at huge indoor car park in a big shopping mall.

Acknowledgement

The author would like to express deeply gratitude to UTHM funded under "Tier 1" Vot No. H262 and also thank to Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for the facilities.

References

[1] S. S. Wang, "A BLE-based pedestrian navigation system for car searching in indoor parking garages," Sensors (Switzerland), vol. 18, no. 5, 2018

[2] F. S. Francis, N. Jali, A. H. Fauzi, and S. K. Jali, "Parkbuddy – Find my car android mobile application," J. Telecommun. Electron. Comput. Eng., vol. 9, no. 3–11, pp. 27–32, 2017

[3] J. Li, Y. An, R. Fei, and H. Wang, "Smartphone based car-searching system for large parking lot," Proc. 2016 IEEE 11th Conf. Ind. Electron. Appl. ICIEA 2016, pp. 1994–1998, 2016

[4] H. C. Tan, J. Zhang, X. C. Ye, H. Z. Li, P. Zhu, and Q. H. Zhao, "Intelligent car-searching system for large park," Proc. 2009 Int. Conf. Mach. Learn. Cybern., vol. 6, no. July, pp. 3134–3138, 2009

[5] S. F. A. Razak, C. L. Liew, C. P. Lee, and K. M. Lim, "Interactive android-based indoor parking lot vehicle locator using QR-code," 2015 IEEE Student Conf. Res. Dev. SCOReD 2015, pp. 261–265, 2015

[6] Z. Chen, J. Fu, R. Sun, and H. Han, "QR code location based reverse car-searching route recommendation model," 2014 IEEE Int. Conf. Veh. Electron. Safety, ICVES 2014, pp. 73–77, 2014

[7] P. C. Lee, S. S. Wang, "A Route Guidance System for Car Finding in Indoor Parking Garages," International Journal of Computer and Information Engineering, 10(10), 1715-1720, 2016

[8] GitHub. 2020. Openalpr. [online] Available at: <https://github.com/openalpr> [Accessed 16 August 2020]

[9] Sensorlink.Sdn.Bhd. [License Plate Recognition], 2017. Retrieved from http://www.sensorlink.com.my/license-plate-recognition/

[10] S.-W. C. Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, "Automatic license plate recognition," Adv. Intell. Syst. Comput., vol. 727, no. 1, pp. 67–77, 2019