

## Development of portable automatic number plate recognition (ANPR) system on Raspberry Pi

S. Fakhar A. G., M. Saad H., A. Fauzan K., R. Affendi H., M. Aidil A.

Faculty of Electrical & Electronic Engineering Technology, Universiti Teknikal Malaysia Melaka (UTeM), Malaysia

---

### Article Info

#### Article history:

Received Aug 11, 2018

Revised Nov 20, 2018

Accepted Dec 11, 2018

---

#### Keywords:

Image processing  
Number recognition  
OpenCV  
Portable ANPR  
Raspberry Pi

---

### ABSTRACT

ANPR system is used in automating access control and security such as identifying stolen cars in real time by installing it to police patrol cars, and detecting vehicles that are overspeeding on highways. However, this technology is still relatively expensive; in November 2014, the Royal Malaysian Police (PDRM) purchased and installed 20 units of ANPR systems in their patrol vehicles costing nearly RM 30 million. In this paper a cheaper alternative of a portable ANPR system running on a Raspberry Pi with OpenCV library is presented. Once the camera captures an image, image desaturation, filtering, segmentation and character recognition is all done on the Raspberry Pi before the extracted number plate is displayed on the LCD and saved to a database. The main challenges in a portable application include crucial need of an efficient code and reduced computational complexity while offering improved flexibility. The performance time is also presented, where the whole process is run with a noticeable 3 seconds delay in getting the final output.

Copyright © 2019 Institute of Advanced Engineering and Science.  
All rights reserved.

---

### Corresponding Author:

Shamsul Fakhar Abd Gani,  
Faculty of Electrical & Electronic Engineering Technology,  
Universiti Teknikal Malaysia Melaka (UTeM),  
Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.  
Email: shamsulfakhar@utem.edu.my

---

## 1. INTRODUCTION

Automatic Number Plate Recognition (ANPR) technology has played a substantial role in automated traffic law enforcement. An ANPR system is essentially a means of identifying a vehicle number plate by extracting the information from an image file using image processing techniques. The process typically consists of image acquisition, image preprocessing, determination and extraction of region of interest (ROI), and interpreting the pixels into numerically readable characters using optical character recognition (OCR) [1]-[3].

There are many techniques used to improve the accuracy of the system; some common ones are RGB, YcbCr, image filter, fuzzy algorithm, image binarization, support vector machines, genetic algorithm and artificial neural network [4], [5]. This paper aims to develop an ANPR system running solely on the Raspberry Pi using OpenCV.

Raspberry Pi is a card-sized cheap mini-computer aimed to make computing accessible to the public. While the original intention of the Raspberry Pi is to provide a base for kids to learn programming, it is also gaining popularity among tech-enthusiasts as it can be used to do different types of commercial programming. It serves as an efficient base due to its low cost and the number of interfaces available. The Raspberry Pi can be used instead of a personal computer, but with some limitations due to its limited processing power [6], [7]. Table 1 further illustrates the specification details of the Pi 3 used in this project.

Table 1. Raspberry Pi 3 Specification Details

Specification	Details
SoC	Broadcom BCM2837
CPU	4× ARM Cortex-A53, 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)
Networking	10/100 Ethernet, 2.4GHz 802.11n wireless
Bluetooth	Bluetooth 4.1 Classic, Bluetooth Low Energy
Storage	External microSD
GPIO	40-pin header, populated
Ports	HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

An 8 megapixels (MP) Raspberry Pi camera can be attached to the on-board Raspberry Pi camera connector, and this creates an image capture system with embedded computing that can extract information from images without the need for an external processing unit. The multiple GPIOs available can interface with external devices and can be used to make results available to other devices. Considering the requirements of image processing compared to the Raspberry Pi's processing module and its peripherals, it is decided that the system is capable on executing the tasks specified. Experimental results show that the designed system is decent enough to run the image capturing and image recognition algorithm [8].

## 2. PROPOSED DESIGN

The overall system design can be categorized into 2 parts, the hardware design and the software design. The hardware design as depicted in Figure 1 will be entirely run on a Raspberry Pi 3 with Raspbian Jessie OS installed, with additional peripherals of an 8 MP camera to capture images, and a 3.5" TFT LCD to display the results. The recognized number plates will also be logged to a cloud database inside Pi. The system is intended to be portable so it is powered by connecting it to a mobile powerbank.

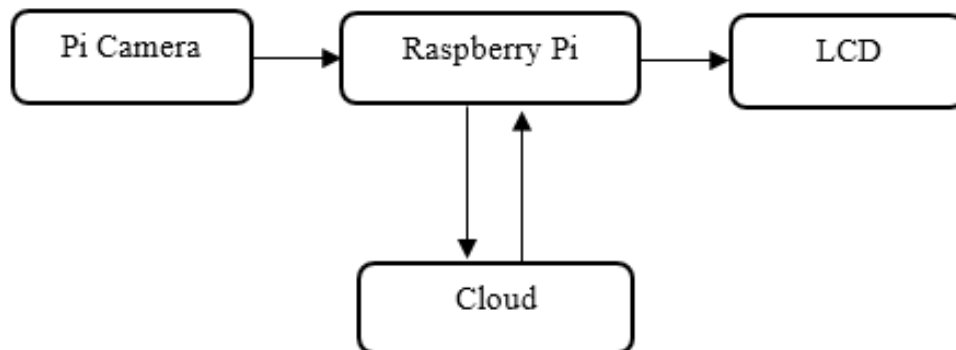


Figure 1. Hardware requirements

In addition to the Python programming language that is natively used in Raspberry Pi environment, this system will also use OpenCV, a computer vision simulation from Intel [9] to help with image pre-processing that is designed with criterias of resource optimization, low power consumption and improved speed. The proposed portable ANPR system software implementation can be categorized into 7 procedural steps as illustrated in Figure 2. Each process is crucial because the result of one particular process will be delivered on to the next process.

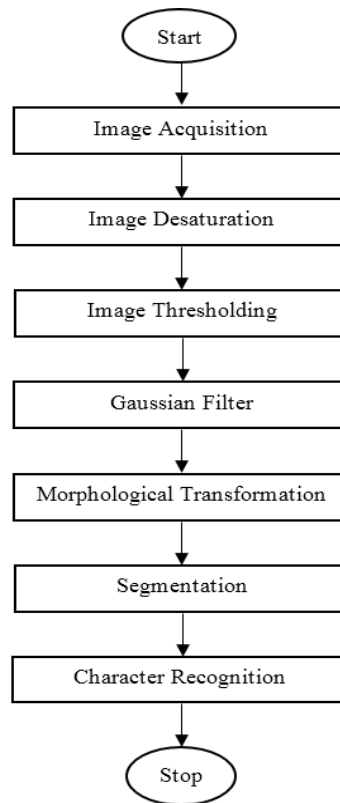


Figure 2. Implementation flowchart

### 2.1. Image acquisition

An 8 MP Pi NoIR camera capable of taking infrared photos up to  $3280 \times 2464$  pixels is used. This camera is also capable of capturing video at 1080p30, 720p60 and  $640 \times 480$ p90 resolutions which is a high quality video [10]. To reduce the computational load on the Raspberry Pi, this camera is setup to capture images with only  $640 \times 480$  pixels. Before testing the system with a live image, a test image is preloaded to test the algorithm as shown in Figure 3.



Figure 3. Loaded test image

### 2.2. Image desaturation

The next step is converting the colour image into grayscale by applying image desaturation [11]. This is done to reduce the complexity of processing colour image. A grayscale digital image is a single sample, carrying only intensity information, and this can facilitate the processing better [12]. OpenCV has the `cvtColor` function to convert color images to grayscale and the effect of this function is demonstrated in Figure 4.



Figure 4. Image covered into grayscale

### 2.3. Image thresholding

Thresholding is the process of converting the grayscale image into a bi-level image. It does not identify main objects, but rather separate them from the background in order to obtain the information that we have to deal with. During thresholding, it is crucial to set the correct threshold value which will best determine a pixel as an object or a background. We use OpenCV's `cvThreshold` to achieve this, and the result is as illustrated in Figure 5.



Figure 5. Image after thresholding

### 2.4. Gaussian filter

Gaussian filter is blurring an image by applying a Gaussian function. It is used to reduce image noise and reduce unwanted detail. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen. Gaussian filter is typically used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales [13]. The effect of applying this function is demonstrated in Figure 6.

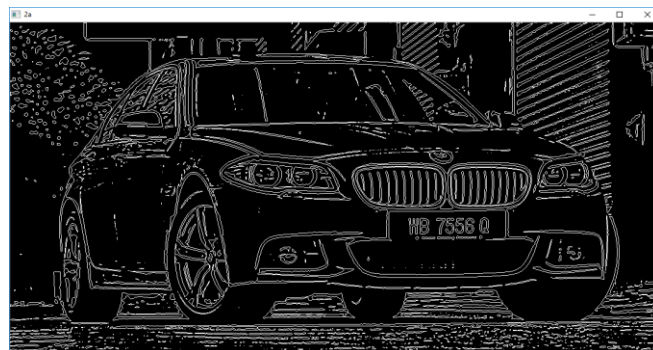


Figure 6. Image after Gaussian filter

## 2.5. Morphological transformation

Morphological transformation inspects geometrical structure within an image by probing it with small patterns called structuring elements. The result is a nonlinear image operator that is well-suited for exploring geometrical and topological structure. The operator is applied to an image in order to make certain features apparent, and distinguish meaningful information from irrelevant distortions by reducing it to a skeleton [14]. This process has 4 kinds of operations: expansion, corrosion, opening and closing operation. Figure 7 shows the effect of the image after morphological transformation.

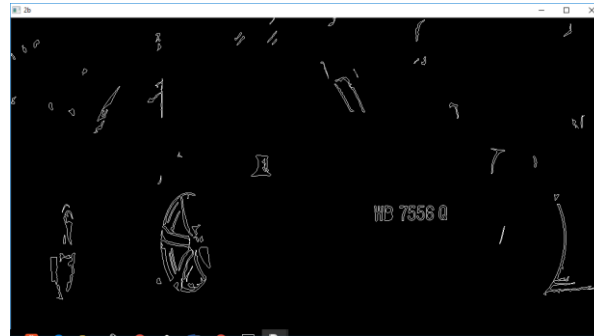


Figure 7. Image after morphological transformation

## 2.6. Segmentation

Image segmentation is the process of dividing an image into multiple parts, typically used to identify objects or other relevant information in digital images [15]. After morphological transformation, 2 segments were identified as potential candidates as depicted in Figure 8. The segments will then go through another process where the system will determine which one has the greatest possible character. The segment will be isolated in a square image, and it is tested so that the longest list of potential character will be determined as the actual number plate. The last image with the most potential characters in it will then be chosen to go for character recognition in the next process.

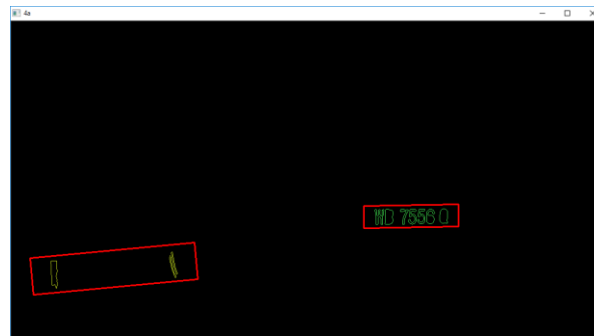


Figure 8. Candidates for recognition highlighted

## 2.7. Character recognition

To recognize alphanumeric characters from the segmented image, the potential number plate is segmented further into individual characters before OCR testing as shown in Figure 9.

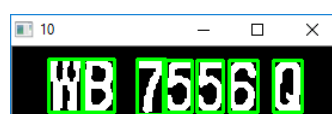


Figure 9. Individual character segmentation

This system uses the K-Nearest Neighbour (KNN) method which requires running a set of test on the character library against the pixels of the separated individual characters from the image [16]. A similarity function  $s()$  can be derived from a distance function  $d()$ . When  $d(p1, p2) \in [0,1]$  we define the similarity as

$$s(p1, p2) = 1 - d(p1, p2)$$

where  $0 \leq s(p1, p2) \leq 1$  for any  $p1$  and  $p2$

The system will determine the output of the alphanumeric number plate by printing it on the image. KNN algorithm operates by giving a set of training data to generate the data structure. For this system, multiple data structure has been generated to get the highest percentage of accuracy of the character recognition [17].

### 3. EXPERIMENTAL RESULTS AND DISCUSSION

The system is then set to capture images from the calibrated [18] Pi NoIR camera and samples of 100 images were captured and tested. It is observed that the system manages to deliver good results when the subject is within 2 meters from the camera. Samples of the successful recognitions image were as illustrated in Figure 10 – 13.



Figure 10. Successful recognition



Figure 11. Successful recognition



Figure 12. Successful recognition



Figure 13. Successful recognition

Sometimes the system fails to recognize characters correctly, where letters are filtered out causing the character not to be registered in the data structure as depicted in Figure 14. In some cases, the system confuses similar letters such as D from O or 6 from 8 and so on. This is illustrated in Figure 15 and ultimately in Table 2.





Figure 14. Eliminated character error



Figure 15. Character mismatch error

Table 2. Common Recognition Errors

Actual	Recognized	Actual	Recognized
B	8, 6	R	P
D	0, O	U	V
G	6	V	U
I	1	W	V
K	X	0	D, Q
M	N	1	I
N	M	3	B
O	0	6	G
P	R	8	B
Q	0		

Based on 100 samples taken, it is found that the processing on the Raspberry Pi 3 took 2 to 3 seconds to process the image and come out with the recognized number plates, with a success recognition rate of 85%.



Figure 16. The complete system

#### 4. CONCLUSION

The development of a portable ANPR system using Raspberry Pi 3 is demonstrated in this article, implementing OpenCV as the core image processing software. The system performs magnificently, but it is hoped that the Raspberry Pi foundation can come out with a newer model with better processing power to eliminate the 3 seconds delay needed to produce the results. In the meantime, future work may be done to improve the accuracy in the recognition process as well as speed up the time taken to produce the output on the algorithm side.

## ACKNOWLEDGEMENTS

The authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM) for providing the support needed to complete the work herein via grant number: RAGS/1/2015/ICT01/FTK/03/B00115 and PJP/2016/PKA/FTK-CACT/S01512.

## REFERENCES

- [1] H. Rajput, T. Som, and S. Kar, "An Automated Vehicle License Plate Recognition System," *Computer (Long Beach, Calif.)*, vol. 48(8), pp. 56–61, 2015.
- [2] S. R. Aher and N. D. Kapale, "Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition," *Int. Res. J. Eng. Technol.*, pp. 2395–56, 2017.
- [3] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23(2), pp. 311–325, 2013.
- [4] Z. X. Chen, C. Y. Liu, F. L. Chang, and G. Y. Wang, "Automatic License-Plate Location and Recognition based on Feature Saliency," *IEEE Trans. Veh. Technol.*, vol. 58(7), pp. 3781–3785, 2009.
- [5] A. Mutholib, T. S. Gunawan, J. Chebil, and M. Kartiwi, "Development of Portable Automatic Number Plate Recognition System on Android Mobile Phone," in *IOP Conference Series: Materials Science and Engineering*, vol. 53(1), 2013.
- [6] D. P. R. D. Ms. Sejal V. Gawande, "Raspberry Pi Technology," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 5, no. 4, pp. 37–40, 2015.
- [7] The MagPi Magazine, "Raspberry Pi 3 is out now! Specs, Benchmarks & More," *MagPi Mag.*, pp. 1–14, 2016.
- [8] Senthilkumar G, Gopalakrishnan K, and Satish Kumar, "Embedded Image Capturing System Using Raspberry Pi System," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 3(2), pp. 213–215, 2014.
- [9] M. Teena Ravali and R. Sai Komaragiri, "Image Processing Platform On Raspberry Pi For Face Recognition," *Glob. J. Adv. Eng. Technol. ISSN*, vol. 3, pp. 2277–6370, 2014.
- [10] W. F. Abaya, J. Basa, M. Sy, A. C. Abad and E. P. Dadios, "Low Cost Smart Security Camera with Night Vision Capability Using Raspberry Pi and OpenCV," *International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, Palawan, pp. 1–6, 2014.
- [11] G. Xie and W. Lu, "Image Edge Detection Based On Opencv," *Int. J. Electron. Electr. Eng.*, vol. 1(2), pp. 104–106, 2013.
- [12] H. Sugano and R. Miyamoto, "Highly Optimized Implementation of OpenCV for the Cell Broadband Engine, " in *Computer Vision and Image Understanding*, vol. 114(11), pp. 1273–1281, 2010.
- [13] Z. Zivkovic, "Improved Adaptive Gaussian Mixture Model for Background Subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol.2, pp. 28–31, 2004.
- [14] H. İ. Çelik, L. C. Dülger, and M. Topalbekiroglu, "Fabric Defect Detection Using Linear Filtering and Morphological Operations," *Indian J. Fibre Text. Res.*, vol. 39(3), 2014.
- [15] N. Singh and M. J. Delwiche, "Machine Vision Methods for Defect Sorting Stonefruit," vol. 37(6), pp. 1989–1997, Nov 1994.
- [16] G. Amato and F. Falchi, "kNN based Image Classification Relying on Local Feature Similarity," *Proc. Third Int. Conf. Similarity Search Appl. - SISAP '10*, pp. 101, 2010.
- [17] R. A. Hamzah, M. S. Hamid, A. F. Kadmin, and S. F. A. Ghani, "Improvement of Stereo Corresponding Algorithm based on Sum of Absolute Differences and Edge Preserving Filter," in *Proceedings of the 2017 IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2017*, pp. 222–225, 2017.
- [18] R. A. Hamzah, S. F. Abd Ghani, A. F. Kadmin and K. A. A. Aziz, "A Practical Method for Camera Calibration in Stereo Vision Mobile Robot Navigation," *IEEE Student Conference on Research and Development (SCORED)*, Pulau Pinang, pp. 104-108, 2012.

## BIOGRAPHIES OF AUTHORS



**Shamsul Fakhar bin Abd Gani** graduated from Universiti Malaysia Perlis (UniMAP) in Bachelor of Engineering (Computer Engineering) with honours in 2006 and later received his Master's degree in Internet & Web Computing in 2015 from Royal Melbourne Institute of Technology (RMIT) Australia. He started his career as an R&D electronic engineer specializing in software design for meter cluster development in Siemens VDO Automotive Penang (later known as Continental Automotive Malaysia). Before leaving Continental, Shamsul plays an active role in Mitsubishi Fuso projects as a software developer and also software project manager. Shamsul is now a lecturer in electronic and computer engineering technology department of FTKEE UTeM.





**Mohd Saad** born in 1981, graduated from Multimedia University (MMU) where he received his B. Eng majoring in Computer in 2003. He then continues working with Telekom Malaysia Berhad and later joined Siemens VDO Automotive Penang (later known as Continental Automotive Malaysia) as a software engineer. He developed firmware in meter cluster development for both local and international customer in the area of Controller Area Network and ECU diagnostic during his tenure in the company. In 2014 he received his M. Eng majoring in Computer and Communication from Universiti Kebangsaan Malaysia (UKM). Currently he is a lecturer in Universiti Teknikal Malaysia Melaka teaching programming, embedded system and microcontroller subjects.



**Ahmad Fauzan bin Kadmin** CEng. P.Tech. currently attached with UTeM as a researcher, Ahmad Fauzan bin Kadmin CEng. P.Tech. has over 14 years of experience in electronic & computer engineering field with technical expertise in R&D engineering, computer vision & medical electronics. He graduated with a Bachelor Degree in Electronic Engineering from Universiti Sains Malaysia (USM) and Master Degree in Computer & Communication Engineering from Universiti Kebangsaan Malaysia (UKM). Previously, he worked with Megasteel Sdn. Bhd., Samsung SDI (M) Sdn. Bhd. and Agensi Angkasa Negara. He published several technical and engineering paperworks in image processing and medical electronics.



**Rostam Affendi Hamzah** graduated from Universiti Teknologi Malaysia (UTM) where he received his B. Eng majoring in Electronic Engineering in 2000. In 2010 he received his M. Sc. majoring in Electronic System Design Engineering from Universiti Sains Malaysia (USM). In 2017, he received PhD majoring in Electronic Imaging from Universiti Sains Malaysia (USM). Currently he is a lecturer in Universiti Teknikal Malaysia Melaka teaching digital electronics and digital image processing.



**Muhammad Aidil bin Azhar** graduated from Universiti Teknikal Malaysia Melaka (UTeM) in Bachelor of Computer Engineering Technology (Computer System) with Honours in 2018. He is currently working as a test and validation engineer in a firmware department in Western Digital (Malaysia).