

PAPER • OPEN ACCESS

Real time mobile based license plate recognition system with neural networks

To cite this article: Connie Liew *et al* 2020 *J. Phys.: Conf. Ser.* **1502** 012032

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

Real time mobile based license plate recognition system with neural networks

Connie Liew¹, Chin Kim On¹, Rayner Alfred¹, Tan Tse Guan², Patricia Anthony³

¹ Knowledge Technology Research Unit, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Jalan UMS, 88400 Kota Kinabalu, Sabah.

² Faculty of Creative Technology & Heritage, Universiti Malaysia Kelantan, Kelantan, Malaysia

³ Faculty of Environment, Society and Design, Lincoln University, Christchurch, New Zealand

kimonchin@ums.edu.my

Abstract. In this paper, the implementation of localizing and recognizing license plate in real time environment with a neural network using a mobile device is described. The neural networks used in this research are Convolutional Neural Network (CNN) and Backpropagation Feed Forward Neural Network (BPFNN). Image processing algorithm for pre-processing, localization and segmentation is chosen based on its ability to cope with limited computational resource in mobile device. The proposed license plate localization steps include combination of Sobel edge detection method and morphological based method. Detected license plate image is segmented using connected component analysis (CCA) and bounding box method. Each cropped character is fed into CNN or BPFNN model for character recognition process. The neural network model was pretrained using desktop computer and then later exported and implemented in Android mobile device. The experiment was conducted in a moving vehicle on selected driving routes. The results obtained showed that CNN performed better compared to BPFNN in a real time environment.

1. Introduction

Automatic License Plate Recognition (ALPR) is an image technology application that detects, extracts and recognizes license plate information from an image or video frame. The extracted information can be used in many applications, such as real time traffic monitoring system, vehicle access control and electronic payment system which include toll payment, parking fee payment and so on [1]. In real life application, ALPR system must cope with various challenges such as different types of license plate, with varying font and color and inconsistent license plate position that increases the difficulty in detecting and recognizing license plate [2]. Conventional ALPR system is usually installed as a fixed-point camera, which is not portable, expensive and requires a high-resolution camera and desktop/notebook computer to process the image using complex image processing algorithm [3]. The accuracy of a license plate detection and recognition using high quality image is often higher than a low-quality image. Fortunately, smartphones nowadays are equipped with high-performance processor and decent camera hardware that can produce high quality image, thus making it possible to implement complex image processing algorithm for ALPR system. Android mobile platform provides infrastructure libraries that allow developer to access the mobile device camera hardware



programmatically. There are many existing open-source computer vision libraries that support Android mobile development such as OpenCV SDK [4].

The implementation of real time ALPR system is complex and challenging. In this research, the moving camera approach is used instead of static camera. Moving camera approach is more complicated than static camera as the background continuously changes and there is no fixed position of where the license plate will be located in the video frame. Moving camera approach for LPR system utilizes real time computer vision techniques where the use of optimized algorithms in real time environment is vital to process instant license plate detection and recognition. The implementation of image processing algorithms is computationally intensive and resource exhaustive. Due to limited CPU and memory resources in mobile device, the implementation of an image processing technique needs to be considered as well. In this research, an image processing algorithm that is less complex and requires less processing time is preferred. For example, morphological based method and Sobel edge detector are suitable due to its simplicity, better performance and less processing time compared to other localization algorithms [5]. Meanwhile, the training process for NN is carried out in a desktop computer instead of a mobile device to speed up the process. There are particular challenges associated with real time LPR in mobile devices. For example, the camera installed onboard might be unstable due to driving in high speed or uneven road surface which may result in the camera losing focus. Driving speed limit needs to be maintained because driving at higher speed will increase the safety distance between car in front and thus increasing the distance between camera and target license plate. This will make it difficult for the target car to get into the detection range of the system and the size of license plate will appear much smaller in the video frame. Other than that, selecting the right camera resolution is important to ensure that a high-quality image is obtained as low quality image will affect the license plate detection accuracy.

In [6], the author proposed real time LPR method for moving camera in a complex scene using the color information of braking light on the vehicle which yielded very good results. However, the method was implemented on a computer, and it consists of several major processing steps such as filtering image background with color information on braking lights and histogram equalization. These methods are not suitable for implementation in mobile device as it requires longer processing time to process color pixels on an image instead of binary image. [7] implemented real time ALPR system for outdoor parking lots using CNN with GPU accelerated function. This method can effectively deploy complex networks model in real time and yielded very good results. However, the method needs to be carried out in a computer equipped with powerful GPU which is not suitable for mobile device implementation.

Artificial Neural Network (ANN) is also commonly referred to as neural networks (NN) is among the popular method for solving LPR problems. Researchers have proven the ANN is superior in image classification and recognition [8][9], gaming development and design [10][11], prediction and estimation [12][13]. BPFNN and CNN are among the widely used NNs in LPR research [14]. BPFNN [15] and CNN [16][17] have all been proven to perform well and demonstrated satisfactory results. This is the main reason why BPFNN and CNN are adopted in this study.

2. Proposed method

The proposed method is divided into four stages, namely the image/video frame preprocessing, license plate localization, license plate segmentation and license plate character recognition.

2.1. Pre-processing

The first step in ALPR system is to acquire images or video frames from a camera. In this research, an Android mobile device is used as a moving camera to capture the LPR process. This stage is a three steps process, which is video to video frames conversion, grayscale conversion and noise reduction. The captured video is converted into video frames before undergoing further processing. Each frame will undergo grayscale conversion as grayscale simplifies the algorithm and reduces computational requirements compared to color images which require more color data to be processed [18]. Next,

median filter is used to reduce noise in the frames due to its ability to preserve edges while reducing substantial amount of noise.

2.2. Localization

The second stage in ALPR system is to detect or extract license plate image from the video frame based on the rectangular shape and boundary of the license plate. The proposed method first converts the grayscale image into binary image using Otsu's binarization method. As license plate are usually designed in rectangular shape, vertical Sobel edge operator is applied on the binary input frame to find all possible rectangles that exist in the video frame. The morphological closing operation is then applied to fill the gaps between the license plate character and to smooth the outer edges.

In order to remove unwanted objects in the video frames such as tree, building and road signboard, the contours is identified, and each contour area is analysed. License plate area is predefined, and contours which are larger or smaller than the area is removed. The remaining contour is connected with a bounding box for license plate verification process. The bounding box is analysed in terms of its height and width. Since the size of a bounding box for license plate usually consists of greater width than height, any bounding box that doesn't fulfil this requirement is eliminated as non-license plate region.

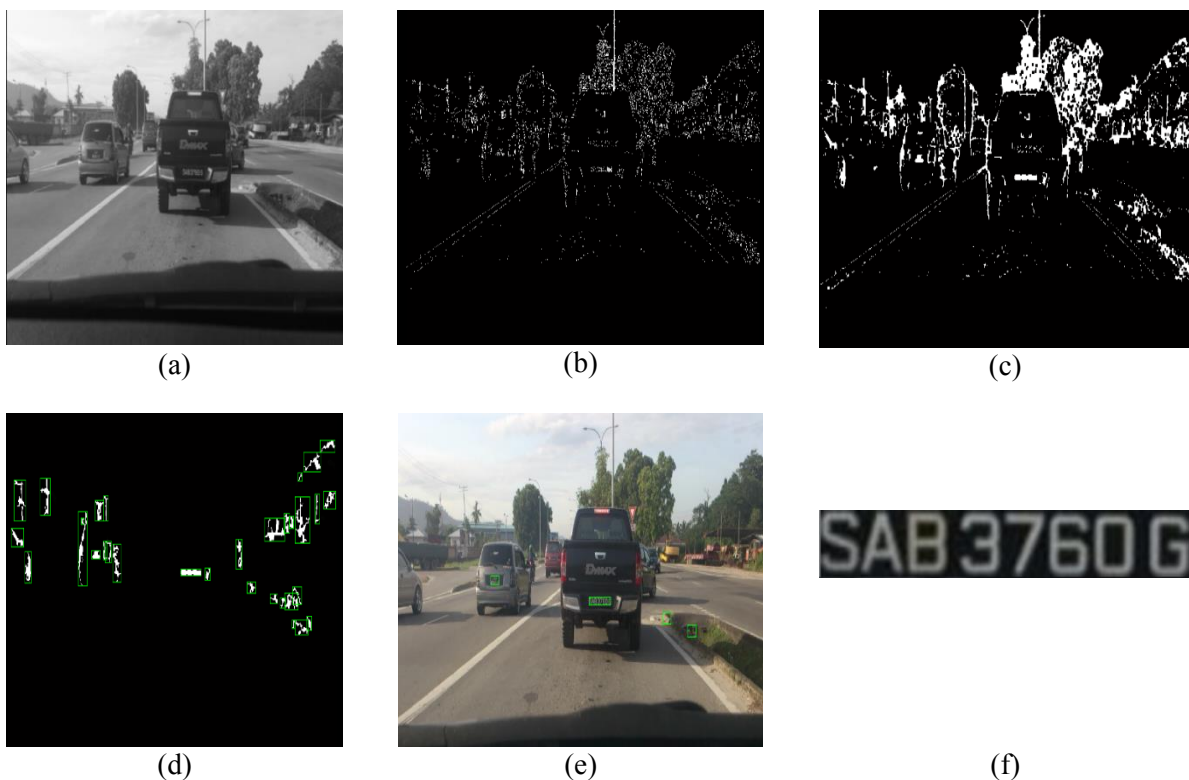


Figure 1. Proposed localization step. (a) Grayscale conversion (b) Vertical Sobel edge detection (c) Morphological closing operation (d) Remaining contour bounded with bounding box (e) Detected license plate region (f) Extracted license plate

2.3. Segmentation

After the license plate is extracted from the video frame, each license plate image is segmented in order to extract and separate each individual character. Binarization method is applied first. Next, connected component analysis (CCA) method is used to find individually connected contour on the

license plate image. CCA analyses and assigns unique label to all connected pixels in the binary image. Each labelled object can be extracted separately according to its label [19]. After that, each separated digit and letter from the license plate image is bounded using bounding box method. Lastly, rectangle region is identified based on the bounded rectangle box, and each bounded character is cropped out as a single input data from the original image. Each cropped out character is normalized into 28x28 images before feeding it into the NN model for character recognition process.

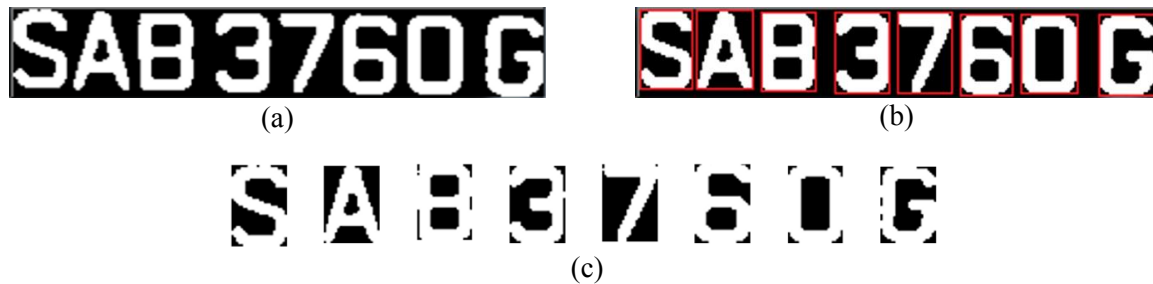


Figure 2. Proposed segmentation step. (a) Binarization (b) Character bounded with bounding box (c) Cropped out character

2.4. Character Recognition

This section describes two algorithms which are CNN and BPFNN. BPFNN is a supervised learning algorithm which uses backpropagation learning method to calculate the gradient of loss in the network [20]. BPFNN propagates through the network to generate output activation. When the forward propagation is accomplished, the output from the forward propagation is compared with the correct output by computing the gradient loss function using gradient descent. The gradient is fed through the network to update the weight value of each hidden neurons. This process continues until the error of the network output is reduced to an acceptable level through certain learning times. The BPFNN model has 784 input neurons, 500 hidden neurons and 62 output neurons with one bias. The NN is trained with datasets from Chars74k [21]. Table 1 shows a summary of BPFNN architecture used.

CNN is a class of deep NN which contains multiple hidden layers and level of basic abstraction [22]. CNN is composed of three different types of layer, mainly convolutional layer, pooling layer, and fully connected layer which can be stacked as [INPUT - CONV - RELU - POOL - FC]. Convolutional layer applies filters on the original image which detects low-level features such as edges and curves. Each filter acts as a feature identifier which slides across all of the areas of the input image and performs multiplication between filter value and original pixel values of the image. Pooling layer performs downsampling operation such as max pooling or average pooling. It is responsible for reducing the dimensionality and control overfitting of the network progressively. Fully connected layer takes the input volume from the output of the preceding convolutional layer or ReLU or pooling layer and generates an output of N-dimensional vector. In this research, the CNN model is a four layers architecture which consists of three convolutional layers and one fully-connected layer. Each convolutional layer uses a 3 by 3 filter size. ReLU is used as the activation function which allows the network to train faster due to its computational efficiency without making significant difference to the accuracy [23]. Max pooling is applied in order to find the maximum value from the filter region of the input volume. The datasets used to train the CNN model is the same as BPFNN. Table 2 shows the summary of CNN architecture used.

Table 1. BPFNN Experimental Settings.

Input Neurons	784
Hidden Neurons	500 + 1 bias
Output Neurons	62 + 1 bias
Learning rate	0.001
MSE	0.01
Function	Sigmoid
Training Samples	46500
Validation Samples	7750
Testing Samples	7750
Maximum No. of Epoch	10000

Table 2. CNN Experimental Setting

Input layer	28x28x1
Convolutional layer #1	8 3x3 filters
Convolutional layer #2	16 3x3 filters
Convolutional layer #3	32 3x3 filters
Activation function	ReLU
Pooling Layer 1	2x2 max pooling
Pooling layer #2	2x2 max pooling
Fully connected layer	62 neurons
Max no. of epoch	1000
Classification output	Softmax
Learning rate	0.001

Tensorflow [24] is a NN framework which provides supports and extensions for mobile platform implementation. The NN model for CNN and BPFNN can be trained on a desktop first instead of a mobile device. The weight matrix and NN structure from the training process is imported into the Android device for character recognition purpose, but without the capability to retrain the weights. This method will save huge amount of the NN training time, and it is more efficient to export the trained NN model directly to the mobile device itself.

3. Experimental results

The LPR process is recorded, and the results are analysed and recorded. Average real time LPR accuracy rate and individual character recognition rate of both CNN and BPFNN are compared.

Table 3. CNN and BPFNN recognition success rate

Type of NN	No. of license plate	No. of fully recognized license plate	No. of partially recognized license plate	No. of license plate not recognized	Average recognition rate
CNN	382	243	65	74	77.20
BPFNN	399	224	83	92	72.96

Based on Table 3, the result is tabulated based on the number of fully recognized license plate, the number of license plate partially recognized and the number of the license plate that is not recognized. A license plate is considered a fully recognized license plate when all the characters on the license plate are recognized correctly, whereas a partially recognized license plate means that only certain characters in the license plate are recognized and finally not recognized license plate indicates a valid license plate that is not recognized at all by the ALPR system. The experimental result shows that the average recognition success rate in CNN is higher compared to BPFNN. CNN was able to fully recognized 64% of the license plates while BPFNN managed to recognise 56% of the license plates. On the other hand, BPFNN was able to partially recognize 21% of license plates as opposed to CNN which only managed to recognize 17% of the license plates. BPFNN was not able to recognize 23% of the license plates whilst CNN was not able to recognize 19% of the license plates. There are several factors that might have contributed to this. Smartphone camera tends to lose focus which may result in the system not being able to recognize license plate that is positioned on left side or the right side of the video frame. Another reason could be due to speed at which the vehicle with the target license plate is moving resulting in a blurred image in the video frame. When driving at a speed above 50

km/h, the minimum safety distance between the vehicle in front will increase which in turn increases the distance of the license plate and the camera greatly. making it difficult to detect the target vehicle license plate and producing a smaller size license plate appearing in the video frame.

Table 4. CNN and BPFNN character analysis

Characters	Recognition success rate (%)	
	CNN	BPFNN
A	96.0	93.7
B	97.3	95.7
C	92.6	92.3
D	92.3	90.9
E	100	100
F	100	100
G	87.5	93.8
H	100	100
J	100	100
K	93.8	95.8
L	82.4	88.9
M	88.2	93.8
N	93.8	93.3
P	100	100
Q	66.7	57.1
R	100	94.7
S	96.6	95.5
T	100	100
U	95.7	100
V	100	100
W	90.6	80.6
X	84.6	81.3
Y	100	86.7
Z	100	100
0	100	98.2
1	93.4	88.6
2	100	98.8
3	95.4	95.1
4	98.1	96.6
5	97.9	97.2
6	97.3	100
7	98.0	98.7
8	98.0	95.4
9	100	100

The character recognition analysis in Table 4 shows that CNN obtains the highest character recognition accuracy rate in the majority of the characters compared to BPFNN. CNN is able to successfully recognized a total of 13 characters which are 'E', 'F', 'H', 'J', 'P', 'R', 'T', 'V', 'Y', 'Z', '0', '2' and '9'. Whereas BPFNN successfully recognized 11 characters which are 'E', 'F', 'H', 'J', 'P', 'T', 'U', 'V', 'Z', '6', and '9'. BPFNN mistakenly categorized more character as other digits and letters compared to CNN. These included characters 'A', 'B', 'D', 'Q', 'S', 'W', 'X', 'Y', '1', '5' and '8'. Whereas CNN wrongly recognized character 'B', 'D', 'Q', 'S', '6' and '8' as other letters and digits. Both NNs often mistakenly recognized some group of similar character such as '8' and 'B', 'A' and '4', 'D' and 'O' or 'D' and '0', 'Q' and 'O', 'S' and 'B' or 'S' and '5', 'Y' and 'T' or 'Y' and 'V'. This happens when the license plate view is tilted or moving further away from the camera or the

license plate font used is not the standard font approved by Malaysia Road Transport Department (JPJ). Among these characters, both NNs had difficulty in recognizing the character 'Q' correctly. The character recognition accuracy rate of character 'Q' is the lowest for both NNs. Figure 3 shows an example of CNN and BPFNN recognition result.

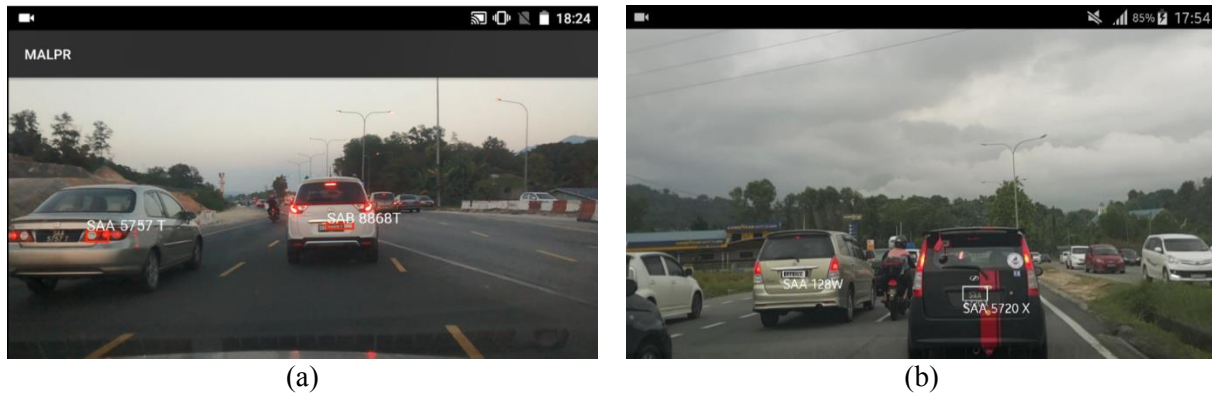


Figure 3. Example of CNN and BPFNN LPR process. (a) CNN ALPR (b) BPFNN ALPR

4. Conclusion and future work

This research investigates the potential of implementing the ALPR system in mobile device for real time environment using a low cost, portable and efficient ALPR system with the use of smartphone device. The ALPR system can be installed in user's smartphone device which is easily accessible by the user. This can reduce the need to carry multiple devices for separate usage. The image processing steps involved are discussed where the image processing algorithm that works best with mobile device is applied. The implementation of both NN in mobile device is discussed as well.

The performance of the proposed CNN and BPFNN was tested based on the video image captured from a moving vehicle on the road. Based on the results obtained, CNN performed better compared to BPFNN in terms of license plate recognition success rate. Both NN models were unable to achieve 100% success rate in recognizing all the characters. This is due to the complex real time environment condition where the system must cope with background changes rapidly, varying distance of license plate and camera, smartphone camera losing focus and motion blur caused by moving vehicles and camera. However, both NN models were able to achieve an average success rate of more than 70%.

The CNN model may be further refined to improve recognition accuracy. The ALPR system can also be tested in varying weather conditions (raining, cloudy or night time in low light condition). In addition, the proposed method can be further refined to improve recognition accuracy while driving with higher speed.

Acknowledgment

This project is partially supported by MESTECC research grant 01-01-10-SF0235.

References

- [1] Rajput H, Som T and Kar S 2015 An automated vehicle license plate recognition system *Computer* vol 48 pp 56-61
- [2] Du S, Ibrahim M, Shehata M and Badawy W 2013 Automatic license plate recognition (ALPR): A state-of-the-art-review *IEEE Trans. Circuits Syst. Video Technol.* vol 23 pp 311-325
- [3] Felix A Y, Jesudoss A and Mayan J A 2017 Entry and exit monitoring using license plate recognition *IEEE Int. Conf. Smart Technol. and Manag. Comput. Commun. Contr. Energy Mater.* pp 227-231
- [4] Bradski G 2000 The OpenCV library *Dr. Dobb's Journal of Software Tools*
- [5] Fradji F, Rezaie A H and Ziaratban M 2007 A morphological-based license plate location *IEEE*

- Int. Conf. Image Process.* pp 57-60
- [6] Hsieh C T, Chang L C, Hung K M and Huang H C 2009 A real time mobile vehicle license plate detection and recognition for vehicle monitoring and management *Joint Conf. Pervasive Comput.* pp 197-202
- [7] Lee S, Son K Y, Yoon B W and Park J 2018 Video based license plate recognition of moving vehicles using convolutional neural network *18th Int. Conf. Contr. Autom. Syst.* pp 1634-1636
- [8] Gan K S, Chin K O, Patricia A and Chang S V 2019 Homogeneous ensemble feedforward neural network in CIMB stock price forecasting *2018 IEEE Int. Conf. Artificial Intell Eng. Technol. (IICAJET)*
- [9] Gan K S, Chin K O, Rayner A, Patricia A, Teo J 2018 A CIMB stock price prediction case study with feedforward neural network and recurrent neural network *J. Telecommunication Electronic and Computer Engineering (JTEC)* **10** 89-94
- [10] Phang W S, Tan L I, Patricia A and Chin K O 2018 Improving the accuracy of stock price prediction using ensemble neural network *Advance Science Letters* **24** 1524-1527
- [11] Ian K W, Chin K O, Rayner A, Ag Ibrahim A A, Tan T G and Patricia A 2018 A review of computer vision methods for fruit recognition *Advance Science Letters* **24** 1538-1542
- [12] Chin K O, Ng W F, Rayner A, Wang C and Tan T G 2018 Development of rule-based procedural content generation for 2-dimensional RPG vol 24 pp 1447-1450
- [13] Ng W F, Chin K O, Rayner A, Ag Asri A I and Teo J 2017 Interactive procedural generation for items in role-playing game *2017 IEEE 2nd Int. Conf. Autom. Control Int. Syst. (I2CACIS)* pp 150-154
- [14] Liew C, Chin K O and Patricia A 2018 A review of automatic license plate recognition system in mobile based platform *J. Telecommunication, Electronic and Computer Engineering (JTEC)* **10** 77-82
- [15] Kim D S and Chien S I 2001 Automatic car license plate extraction using modified generalized symmetry transform and image warping *IEEE Int. Symp. Ind. Electron. Proc.* vol 3 pp 2022-2027
- [16] Dhar P, Guha S, Biswas T and Abedin M Z, 2018 A System Design for License Plate Recognition by Using Edge Detection and Convolution Neural Network *Int. Conf. Comput. Commun. Chemical Mater. Electro. Eng.* pp 1-4
- [17] Izidio D M F, Ferreira A P A and Barros E N S 2018 An Embedded Automatic License Plate Recognition System Using Deep Learning *Brazilian Symp. Comput. Syst. Eng.* pp 38-45
- [18] Kanan C and Cottrell G 2012 Color-to-Grayscale: Does the Method Matter in Image Recognition? *PloS ONE* vol 7
- [19] Chang S L, Chen L S, Chung Y C and Chen S W 2004 Automatic license plate recognition *2004 IEEE Trans. Intell. Transp. Syst.* vol 5 pp 42-53
- [20] Chin K O, Teo K Y, Rayner A, Teo J, Patricia A and Wang C 2016 Backpropagation neural ensemble for localizing and recognizing non-standardized Malaysia's car plate *Int. J. Adv. Sci. Eng. Inf. Technol* **6** 1112-19
- [21] de Campos T E, Babu B R and Varma M 2009 Character recognition in natural images *Proc. Int. Conf. Comp. Vision Theory Applic.*
- [22] Pham V H, Dinh P Q, Nguyen V H 2018 CNN-based character recognition for license plate *Int. Inf. Database Syst. ACIIDS 2018* vol 10752 pp 594-603
- [23] Nair V and Hinton G E 2010 Rectified linear units improve restricted boltzmann machines *Proc. 27th Int. Conf. Mach. Learn.* vol 3 pp 807-814
- [24] Martin A et al. 2015 Tensorflow large-scale machine learning on heterogeneous systems <https://tensorflow.org>