

the license plates. Some car owners also prefer to use different font sizes and font types for their car and this can also complicate the car plate recognition process.

Since the last decade, many approaches have been proposed to localize car plate such as edge detection [5], morphology-based method [6], wavelet transform [7], colour extraction [8], and horizontal and vertical analysis [8], to name a few. In some cases, thinning algorithm was used in the pre-processing for the enhanced images [9]-[11]. The Omhining rough set theory [12], self-organizing feature map (SOFM) neural network [12], novel character classification [10], modified Hausdorff distance template matching [13] and another hybrid method [14], were used to recognize extracted characters from license plates. However, none of the algorithms have been tested for Malaysia license plates thus far. [11], [15] proposed stroke extraction and analysis, and pixel compactness, angles, and projection histogram [15] to localize and recognize car plates. However, the proposed method had to manually localize and detect the characters.

In this research, we propose image deblurring technique to automatically locate the alphanumeric characters from the number plate instead of manually searching the region of the plate. This technique can solve the car plate size variation and the different font types and sizes. Then, a thinning algorithm is applied to reduce the weightage of the pixels without removing any important features. Furthermore, Backpropagation Feed-Forward Neural Network (BPNN) is used to perform character recognition. The result is then compared to the Ensemble BPNN (ENN).

The remainder of this paper is organized as follows. In Section II, the difference between BPNN and ENN is discussed. This section also describes the pre-processing steps in localizing the license plate and the experimental setup. Results and discussions are discussed in Sections III. And finally, the conclusion and future work are described in Section IV.

II. MATERIAL AND METHOD

The proposed method is divided into four stages namely the image pre-processing, the license plate localization, the character segmentation and finally the character recognition step.

A. Image Pre-Processing

This is a two steps process: image binarization and noise filtering. The main objective of this process is to remove most of the noises or unwanted object in the captured images. In the image binarization, the images are converted into binary images that consist of black and white colours. The license plate images are turned into a binary image with 0.7 threshold value. Those values above the threshold are classified as white, otherwise, it is classified as black. Then in the noise filtering process, any detected objects with less than 30 pixels coverage are removed automatically. The noise filtering removed most of the unwanted small objects such as dirt or leaves. Fig. 1 shows samples of the captured images Fig. 2 shows the binarized images and Fig. 3 shows the filtered images.



(a) Normal license plate



(b) Special event license plate



(c) Commemorative license plate

Fig. 1 Captured images



(a) Normal license plate



(b) Special event license plate



(c) Commemorative license plate

Fig. 2 Binarized images

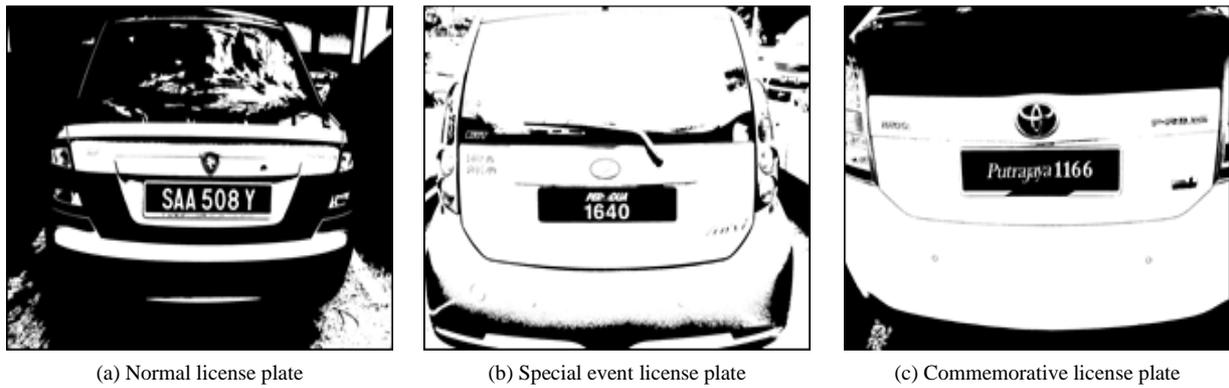


Fig. 3 Noise removed images

Fig. 1 shows three different categories of images taken using a digital camera. The images consist of 1080 HD resolution and saved in jpeg format in order to reduce the storage whilst still maintain the quality. Fig. 1(a) shows one of the standard Malaysia license plates which consists of a combination of alphabets and numbers. Fig. 1(b) shows a special event/commercial license plate with the words “PERODUA” and the number “1640”. The most challenging part of this license plate is the symbol “O” that is not an exactly an “O”, but rather a symbol that looks similar to “O”. Fig. 1(c) shows a commemorative license plate with the words “Putrajaya 1166”, with a capital letter “P. The words “Putrajaya” is in italic format.

B. License Plate Localization

The license plate localization process is quite challenging as a light reflection on the license plate surface, car logo, the name of the car, and other objects may increase the difficulty in localizing the license plate. The proposed method first removed the large objects from the processed images. It removed structure of the car and any objects surrounding the car. However, the algorithm sometimes failed to remove some of the objects in certain cases as shown in the Fig. 4 and Fig. 5. Fig. 4 shows unwanted objects and Fig. 5 shows the results after the removal of large objects.

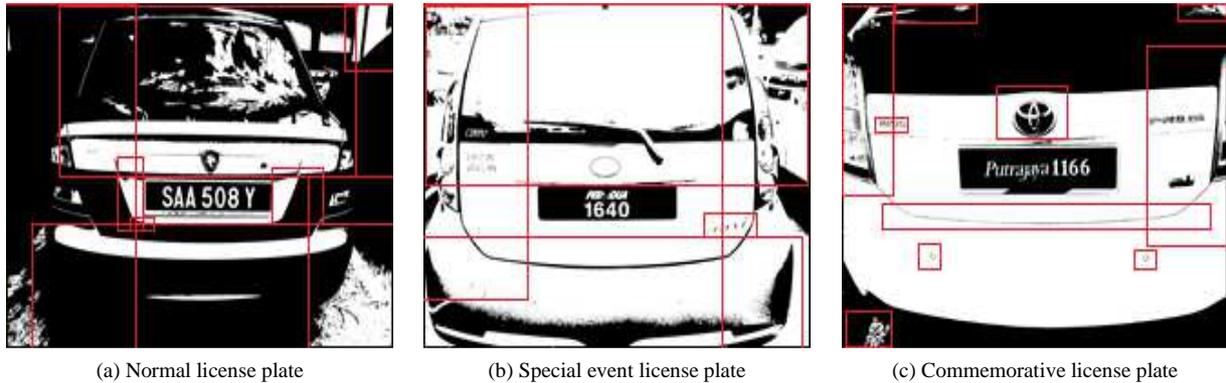


Fig. 4 Unwanted objects (objects surrounded with red box)

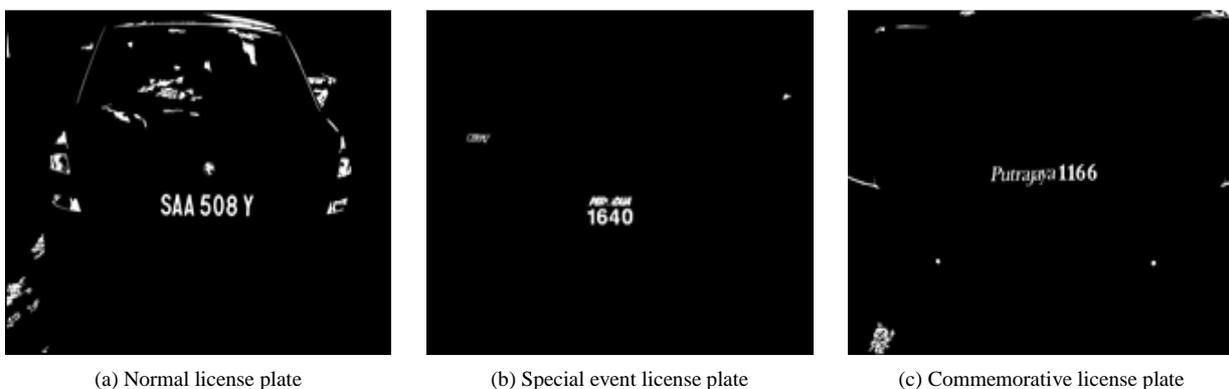


Fig. 5 Large objects have been removed successfully

Fig. 5(a) shows that there are some objects that were not removed after implementing the noise filtering algorithm. These objects are not large objects but they are a combination of small objects. Hence, they remained unchanged in the filtered images. The images are deblurred after the filtering process. The purpose of deblurring is to

create an area so that the letters can be bounded by the bounding box as one object. Then, these unwanted objects will then be removed using the filtering process again. Fig. 6 shows the deblurred images and Fig. 7 shows the bounding box applied to the images.

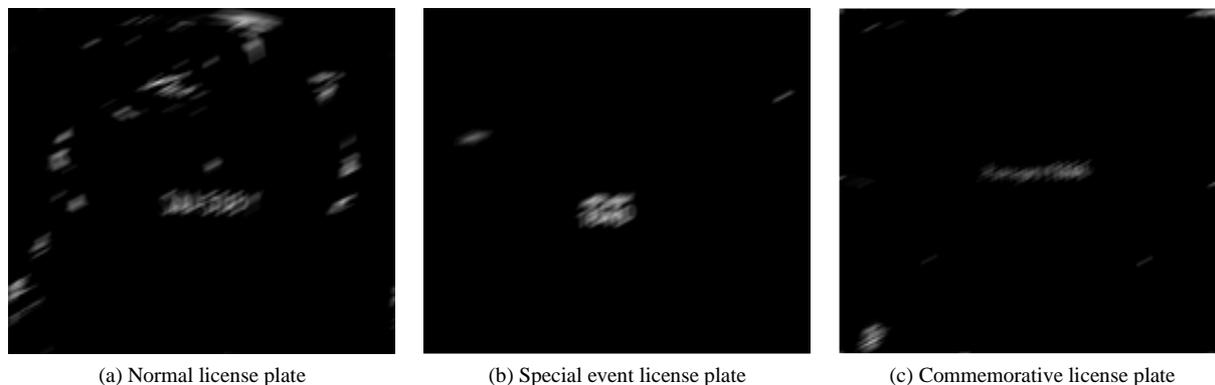


Fig. 6 Deblurred images

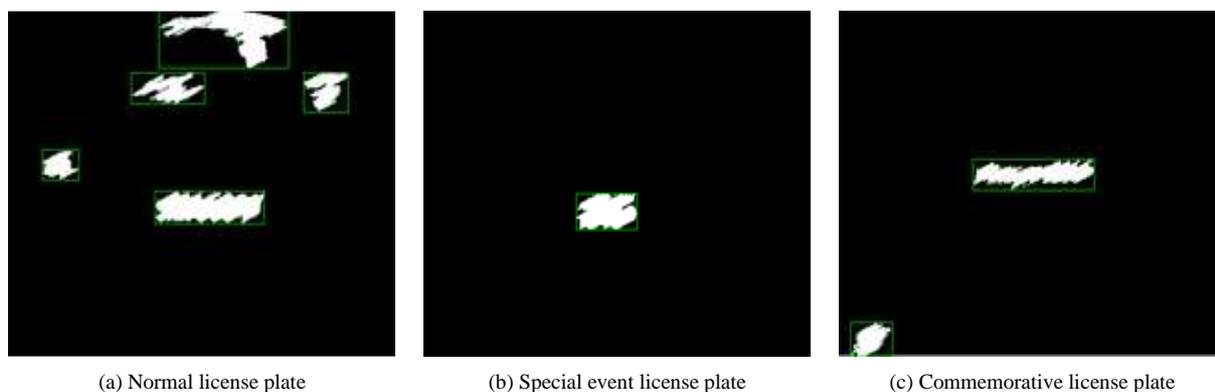


Fig. 7 Bounding objects

Fig. 6 and Fig. 7 show most of the small objects have been combined to become a large object after the deblurring process. The filtering process is again applied in order to remove these unwanted objects. Then, the remained objects are bounded in order to compare its properties. The last step of localization involves the creation of a small bounding box and properties comparison. This is to check whether the bounded object consists of numbers or alphabets or other unwanted objects. This can be validated by comparing the

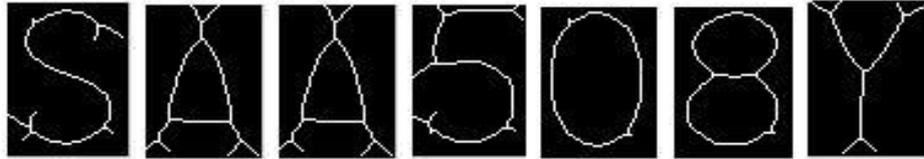
properties of characters and the properties of the bounded objects. Initially, most of the bounded objects are almost similar in size and properties. If there is a match, then the bounded object is assumed to be a license plate. Otherwise, it could be the car's logo or other unwanted objects. The bounded object is then extracted from the images and saved separately. Fig. 8 shows the extracted license plates.



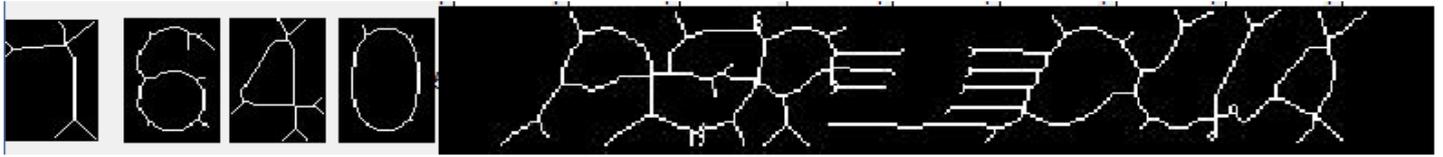
Fig. 8 Extracted license plates

C. Character Segmentation

Fig. 8 shows all the letters are bounded successfully with a red bounding box. Every single bounded object is selected and compared its sizes. If all the bounded objects are similar in size, then this license plate is considered as a normal license plate. Otherwise, it is either a commemorative license plate or a special event license plate. Each detected object is then extracted and saved in a file for thinning.



(a) Normal license plate



(b) Special event license plate



(c) Commemorative license plate

Fig. 9 Thinned images

D. Character Recognition

This section describes two algorithms: (1) Backpropagation Feed-forward Neural Network (BPNN) and (2) Backpropagation Ensemble Feed-forward Neural Network (ENN). BPNN is a supervised learning algorithm used to find an objective function that best scales a set of inputs to its correct output [17]. The BPNN algorithm calculates the gradient of a loss function in the network. The gradient is saved as weight in the network. The gradient is then fed to the neural network which is then used to update the BPNN weights. The updated process is to minimize the loss function. BPNN is a well-known technique and is considered as supervised learning algorithm as it requires a set of input and output values in order to calculate the gradient of the loss function. BPNN can be divided into two phases. The first phase refers to the propagation of the neurons and the second phase involves weight update. In the first phase, the training input data first forward propagates through the NN in order to generate output activations. Next, the output activation backward propagates through the NN using the generated training pattern target. The deltas of all output and hidden neurons are generated during the backward propagation. The second phase of the training involves multiplying the generated output delta and input activation to obtain the gradient of the weight. Then, the process continues with a subtraction of a ratio of the gradient of the weight. The ratio is also named as learning rate and it

The thinning algorithm is used to remove selected foreground pixels from the binary images. It maintains the topology of the original region whilst removing most of the unwanted foreground pixels. Usually, this algorithm reduces all lines to a single pixel thickness for better data representation and used for further processing. We use a conventional thinning algorithm proposed by [16]. Detail of the thinning algorithm could be obtained from [16]. The thinned images are then saved separately for NN training. Fig. 9 shows the thinned images.

influences the quality and speed of the network. The greater the learning rate, the faster the network trains; the lower the learning rate, the more accurate the training is [17]. Phases 1 and 2 are repeated until the performance of the network is satisfied.

In this research, the BPNN involves a single layer feed-forward NN. It consists of one input layer, one hidden layer, and one output layer. Such a simple network is easy to design and have been proven to be efficient and highly effective [18]-[19]. Fig. 10 below shows the BPNN's process.

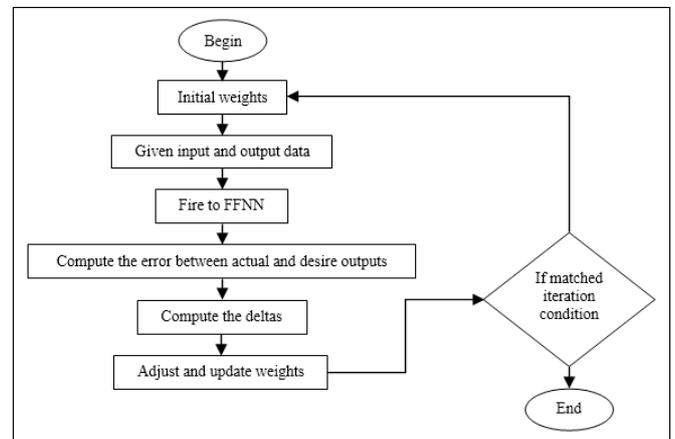


Fig. 10 The BPNN's process

The ENN is similar to BPNN and it is also a learning paradigm but it is a combination of more than one neural networks that are used to solve a single problem. It is a process of joining multiple NN models to produce the desired output. Usually, an ensemble model has better performance than any individual model because the adjustment and update of weights are based on averaged output [20]. Ensemble a number of neural network means training many neural networks and then combining their predictions and then make the decision based on the averaged output. Due to the remarkably good results performed by the ensemble neural network, it has become a very hot topic in both neural network and machine learning communities [21]. Nevertheless, the training for ENN is costly compared to a single layer NN. In this research, the ENN used is a combination of two FFNN models. The outputs of the NN is based on the averaged output from both FFNN models. Weights are assigned to minimize the mean square error of the ensemble. The mean square error is calculated separately with the dataset. Then, the weights are updated according to the averaged mean square error. The ENN formula is represented as below.

$$ENN = \left(\sum_{i=2}^j \omega_{ij} \right) / j \quad (1)$$

where i and j represent the number of NN involved. The ω represents the output of the NN.

The BPNN consists of one input layer, one hidden layer, and one output layer. The input layer consists of 35 neurons, the hidden layer has 52 neurons + 1 bias, and the output layer consists of 43 neurons + 1 bias. The goal of the neural network is set as 0.01 whilst the learning rate of the NN is 0.001. High learning rate could result in an unstable network and in turn oscillate the result. Besides that, the neural network epoch is set to 2000 so that the mean square error

can be dropped to the minimum. The binary sigmoid function is used during the training. The training samples used 200 images w the testing samples used 300 images. Biases were included in both the input and hidden layers in order to finely tune the weights of the NN. Table 1 shows the summary of the experimental setting used in this research.

TABLE I
EXPERIMENTAL SETTING USED FOR BPNN AND ENN

	BPNN	ENN
Input Neurons:	35	35
Hidden Neurons:	52 + 1 bias	52 + 1 bias
Output Neurons:	43 + 1 bias	43 + 1 bias
Learning rate:	0.001	0.001
Goal of Mean Square error:	0.01	0.01
Function:	$1/(1 + e^{-x})$	$1/(1 + e^{-x})$
Training Samples:	200	200
Testing Samples:	300	300
Maximum Number of Epoch:	2000	2000
Bias:	1.0,1.0	1.0,1.0
Number of NN	One FFNN	Two FFNN
ENN Method	-	Averaged output

III. RESULTS AND DISCUSSION

A. License Plate Localization

For license plate localization, the proposed approach achieved 99% success rate using 300 samples. These samples consist of 85% normal license plates, 10% of Malaysia Taxi license plates, and 5% of commemorates license plates. The quality of the sample images is a major factor that caused the failed results. Fig. 11 shows some two examples of failed results.

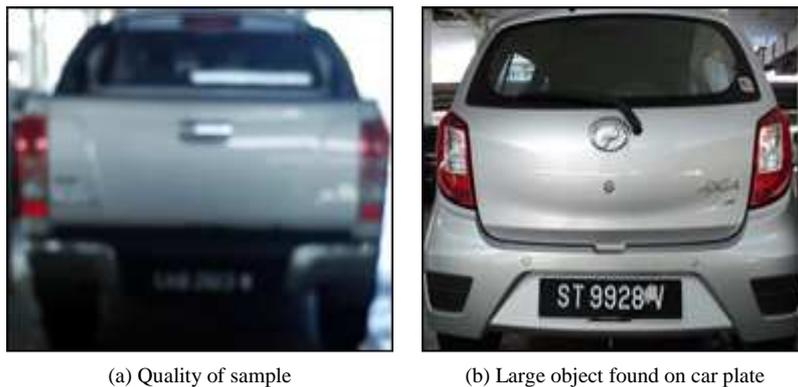


Fig. 11 Failed results in localizing the license plates

B. Character Recognition

The BPNN testing result showed the network model was able to recognize the majority of the characters, with 100% recognition rates. However, the BPNN model only achieved

91% success rate for character “J”, 86% for character “M”, and 97% for both characters “7”, and “8”. The training result also showed that BPNN learnt very fast in the early stage of the optimization process. Fig. 12 shows the mean square

error was reduced significantly during the first 1000 epochs. The NN model was able to achieve the goal at 9233 epochs.

The ENN testing result performed slightly better compared to the BPNN testing result. It achieved 97% of success rate for character “S”, 98% for character “3” and 100% for other characters. The training result shows in Fig. 13 represents that ENN learnt slightly faster than BPNN. The ENN training was completed with 9121 epochs and the BPNN training completed with 9927 epochs. The mean square error was reduced to 0.0618 at epoch 1210, and 0.0277 at epoch 1712. It was able to achieve the goal at epoch 8768. However, the model took 7000 epochs to reduce from 0.0277 mean square error to 0.1012.

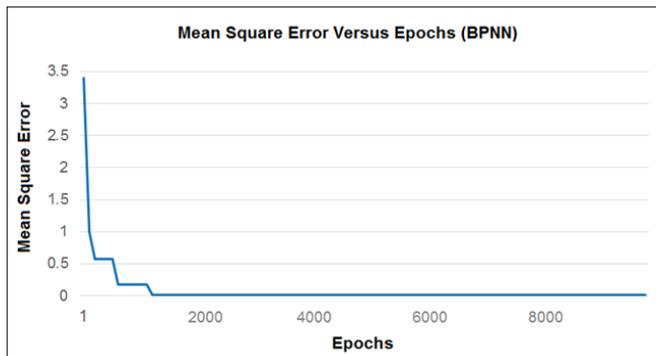


Fig. 12 BPNN training result

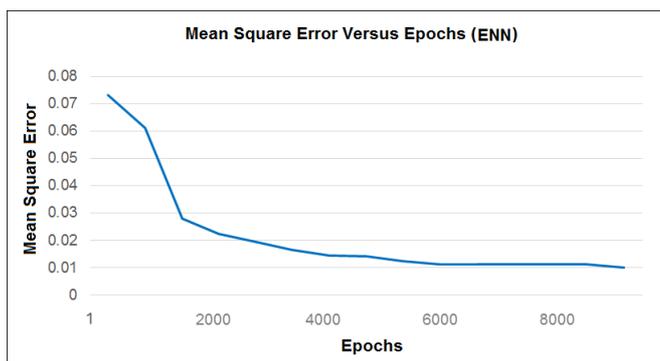


Fig. 13 ENN training result

IV. CONCLUSION

In this paper, we proposed pre-processing steps for localizing license plates. The localization process is simple as it involves only a few steps: binarization, noise filtering, deblur, object filtering, bounding box, and detection. The result is quite promising as the localization success rate achieved 99%.

We also compared the performance of the BPNN and ENN in the training and testing phases. The ENN testing result performed slightly better compared to the BPNN testing result. The ENN model was not able to achieve a 100% success rate in recognizing the characters “S” and “3”. On the other hand, BPNN was not able to achieve a 100% success rate in recognizing the characters “J”, “M”, “7”, and “8”.

In the future, we plan to increase the sample size to 1000 instead of 300, with particular attention to the commemorative license plate. The current approach analyses still image only.

In the future, we intend to test our proposed approach with moving images. We will also perform further experimentation on images that are captured from various angles as well as images that are taken when it is raining or foggy.

ACKNOWLEDGMENT

This research work is partly supported and funded by Universiti Malaysia Sabah, Artificial Intelligence Research Unit (AiRU).

REFERENCES

- [1] JPJ. Jabatan Pengangkutan Jalan Malaysia. (2015) Malaysian Road Transport Department. [Online]. Available: <http://www.jpj.gov.my/web/eng/specifications-of-vehicle-number-plates>
- [2] Thestar.com.my. (2015) Businessman Buys Regal Plate Number. [Online]. Available: <http://www.thestar.com.my/News/Community/2014/09/09/Businessman-buys-regal-plate-number/>
- [3] G1M. (2015) No Plat Gagasan 1 Malaysia (G1M). [Online]. Available: <http://www.g1m.com.my/>
- [4] Lim. A. (2015) IM4U number plate launched, on sale March 10. [Online]. Available: <http://paultan.org/2013/03/04/im4u-registration-plate-launched-on-sale-march-10/>
- [5] P. Rattanathammawat, and T. H. Chalidabhongse, “A car plate detector using edge information,” in *Communications and Information Technologies, International Symposium on IEEE*, 2006, pp. 1039-1043.
- [6] C. Nelson, K. Babu, K. Nallaperumal, “A license plate localization using morphology and recognition,” in *Annual IEEE India Conference, INDICON*, 2008, pp. 34 – 39.
- [7] W. Wen, X. Huang, L. Yang, Z. Yang, and P. Zhang, “Vehicle license plate location method based-on wavelet transform,” in *Computational Sciences and Optimization, CSO 2009, International Joint Conference on IEEE*, 2009, Vol. 2, pp. 381-384.
- [8] E. R. Lee, P. K. Kim, and H. J. Kim, “Automatic recognition of a car license plate using color image processing,” in *Image Processing, ICIP-94., IEEE International Conference*, 1994, Vol. 2, pp. 301-305.
- [9] N. A. Jusoh, J. M. Zain, and T. A. A. Kadir, “Enhancing thinning method for Malaysian car plates recognition,” in *Innovative Computing, Information and Control, ICICIC'07, Second International Conference on IEEE*, 2007, pp. 378-378.
- [10] B. L. Lim, W. Yeo, K. Y. Tan, and C. Y. Teo, “A novel DSP based real-time character classification and recognition algorithm for car plate detection and recognition,” in *Signal Processing Proceedings, ICSP'98. 1998 Fourth International Conference on IEEE*, 1998, Vol. 2, pp. 1269-1272.
- [11] S. Setumin, U. U. Sheikh, and S.A.R. Abu-Bakar, “Car plate character extraction and recognition using stroke analysis,” in *Signal-Image Technology and Internet-Based Systems (SITIS)*, 2010 Sixth International Conference, 2010, pp. 30 – 34.
- [12] M. Fang, C. Liang, and X. Zhao, “A method based on rough set and SOFM neural network for the car's plate character recognition,” in *Intelligent Control and Automation, WCICA 2004, Fifth World Congress on IEEE*, 2004, Vol. 5, pp. 4037-4040.
- [13] Z. You-qing, and L. Cui-hua, “A recognition method of car license plate characters based on template matching using modified hausdorff distance,” in *Computer, Mechatronics, Control and Electronic Engineering (CMCE), 2010 International Conference on IEEE*, 2010, Vol. 6, pp. 25-28.
- [14] P. Xiang, Y. XiuZi, and Z. Sanyuan, “A hybrid method for robust car plate character recognition,” in *Systems, Man and Cybernetics, 2004 IEEE International Conference*, 2004, Vol. 5, pp. 4733 – 4737.
- [15] S. Setumin, U. U. Sheikh, and S.A.R. Abu-Bakar, “Character-based car plate detection and localization,” in *IEEE Computer Vision, Video and Image Processing*, Universiti Teknologi Malaysia, 2010, pp.737-740.
- [16] K.-O. Chin, M. P. Paulraj, and S. Azali, “Certain improvement in pixel based thinning algorithm using neural network,” in *4th*

- National Technical Postgraduate Symposium (TECHPOS 2006)*, 2006.
- [17] J. Heaton. *Introduction to neural networks with Java*, Heaton Research, Inc., 2008.
- [18] K.-T. Chang, K.-O. Chin, J. Teo, and A. M. Jilui-Kiring, "evolving neural controllers using GA for warcraft 3-real time strategy game," in *IEEE The Sixth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2011)*, 2011.
- [19] J. L. Shi, T.G. Tan, J. Teo, K. O. Chin, R. Alfred, and P. Anthony, "Evolving controllers for simulated car racing using differential evolution," *Asia-Pacific Journal of Information Technology and Multimedia*, vol. 2, no. 1, pp. 57-68, 2013.
- [20] Z. H., Zhou, and S. F. Chen, "Neural network ensemble," *Chinese Journal of Computers-Chinese Edition*, vol. 25, no. 1, pp. 1-8, 2002.
- [21] A. J. Sharkey, *Combining artificial neural nets: ensemble and modular multi-net systems*, New York: Springer-Verlag, 1999.