❒   2196

A design of license plate recognition system using convolutional neural network

P. Marzuki, A. R. Syafeeza, Y. C. Wong, N. A. Hamid, A. Nur Alisa, M. M. Ibrahim
Department of Computer Engineering, Technical University of Malaysia Malacca, Malaysia

Article Info	ABSTRACT
	This paper proposes an improved Convolutional Neural Network (CNN) algorithm approach for license plate recognition system. The main contribution of this work is on the methodology to determine the best model for four-layered CNN architecture that has been used as the recognition method. This is achieved by validating the best parameters of the enhanced Stochastic Diagonal Levenberg Marquardt (SDLM) learning algorithm and network size of CNN. Several preprocessing algorithms such as Sobel operator edge detection, morphological operation and connected component analysis have been used to localize the license plate, isolate and segment the characters respectively before feeding the input to CNN. It is found that the proposed model is superior when subjected to multi-scaling and variations of input patterns. As a result, the license plate preprocessing stage achieved 74.7% accuracy and CNN recognition stage achieved 94.6% accuracy.

Corresponding Author:

A. R. Syafeeza,
Department of Computer Engineering,
Technical University of Malaysia Malacca,
76100 Durian Tunggal, Malacca, Malaysia.
Email: syafeeza@utem.edu.my

1. INTRODUCTION

License plate recognition (LPR) system [1]-[3] has been extensively utilized as a part of real life applications such as criminal pursuit, automatic toll collection [4] and enhancing of Automated Enforcement System (AES) performance which aims to control traffic efficiency. In terms of security, LPR has been used in traffic management to detect the owner of the car who has breached the traffic laws and to find stolen vehicles. LPR system is also used for access control to enter a building. The Automatic LPR system was introduced in 1979 at the Police Scientific Development Branch in United Kingdom for security purposes.

Image processing is the main technique to be used in LPR recognition system. Developing the LPR system using image processing is challenging due to limited ability to deal with multi-scaling since the LPR image can appear to be dirty, motion blur, poor resolution, poor lighting, low contrast and etc. The license plate can also appear to be dirty and motion blur. There are five primary stages to identify a license plate. Initially, localization technique is used to find and isolate the license plate on the input image. This is followed by plate orientation to compensate the skew condition of the plate and resizing to adjust the dimensions to the required size respectively. Then, image normalization will be performed to adjust the brightness and contrast of the image. The character segmentation intends to segregate individual character from the license plate.

The recognition part of the LPR system has almost a routine algorithm. It involves adaptive thresholding, component labeling, feature extraction and classification. Among the five major parts, the character recognition process is the most challenging part. This is because, the recognition of the characters is highly dependent on the type of algorithms applied in the first four major parts. In fact, the segmented

characters can appear in various looks. Therefore, a robust character recognition method is required and CNN has possibility solve to the mentioned challenges.

CNN is well-known as a scale and rotation invariant in pattern recognition tasks. CNN accepts raw images that have been preprocessed with the minimal preprocessing algorithm and train the input samples in supervised mode. It combines compression (dimensionality reduction), feature extraction and classification processes in a single architecture. Until now, CNN has been applied to various applications such as face detection [5]-[10], face recognition [11]-[15], gender recognition [16]-[19], object classification and recognition [20]-[22], character recognition [23]-[25], texture recognition [26], finger-vein [27], etc. Despite the listed advantages, CNN has limitations in terms of cost and speed. This is due to the compute intensive image processing algorithm being incorporated in the design such as convolution and subsampling. The convolution process takes almost 90% of the processing time [28]. Therefore, in order to overcome the limitation, designing a small CNN size could aid in reducing the processing time.

The LPR using CNN has been reported in [29]. However, the characters are manually segmented while the real problem of LPR started from the preprocessing stage. In [30], they implemented LeNet-5 architecture with 7 layers by inserting the whole license plate as input and reported 98.25% accuracy. This work classifies between the license plate and non-license plate and not recognizing the characters. They used 2400 license plates and 4000 non-license plate dataset and divided into train and test dataset. Besides, the accuracy rate on license plate detection is incomparable with this work have shown that system performed at the other researches improved on the preprocessing part to improve the result obtained on recognition.

In [31], they proposed two local binary methods, which are local Otsu and an improved Bernsen algorithm. Connected Component Analysis (CCA) is used for binary images searching in an eight-connectivity situation. Besides, according to [32], the labelling algorithm uses a '4-connectivity' method to mark the group of connected pixels and labels them using different numbers. For the recognition part [33], used template matching and achieved the accuracy of 87%. Based on [34], the character region is calculated by using variance projection algorithm. This is used to enhance its noise immunity and improve the segmentation accuracy. An iterative mean filter is used to smooth the original vertical variance projection graph in order to find the corresponding peak to determine the number of character in the license plate. The accuracy achieved unsatisfied.

2. METHODOLOGY
2.1. Database collection
In fact, the license plate dataset are difficult to obtain since their privacy concern. Therefore the images of vehicle license plate are randomly captured around Malacca area as datasets. The images are set up as RGB, 256 bit with 1280x800 resolution. The dataset taken exceed 1000 of RGB images. 700 of 1000 are used as training and the remaining as testing dataset.

2.2. LPR system flowchart
According to Figure 1, it is illustrated the flow chart of overall LPR system phases. The detail algorithm of enhanced SDLM can be referred in [35]. MATLAB and C language has been used as the platform. The overall system consists of three main phases: Preprocessing, Segmentation and Character Recognition. The uniqueness of this approach compared to other existing works on Malaysian's license plate is the implementation of CNN at the recognition part. The whole methodology will be explained in next sections.
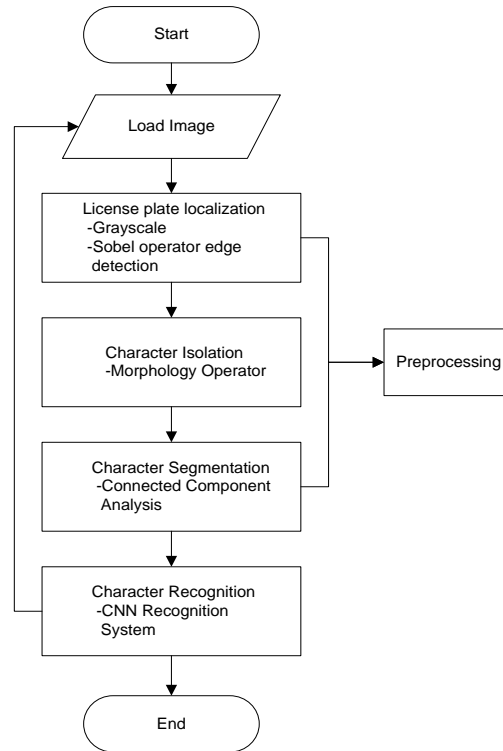
Figure 1. Flow chart of LPR system

2.3. Preprocessing

Preprocessing is the initial stage of image processing tasks to enhance the quality of the image. In this stage, noises are reduced and unwanted features are eliminated to ease the burden of the CNN at the recognition stage. The preprocessing steps involve for LPR includes the following sequence.

2.3.1. License plate localization

The captured images are in RGB format. The images are converted to grayscale to ease the computational process (Figure 2). After that, the grayscale images are processed by Sobel operator edge detection. A threshold of the edge detection is set in order to decrease the sensitivity. By doing this, the edges that are not significant will be ignored. Two histogram graphs are produced based on the edge detection process. The histogram graph represents the column-wise and row-wise histogram of the image as shown in Figure 3.

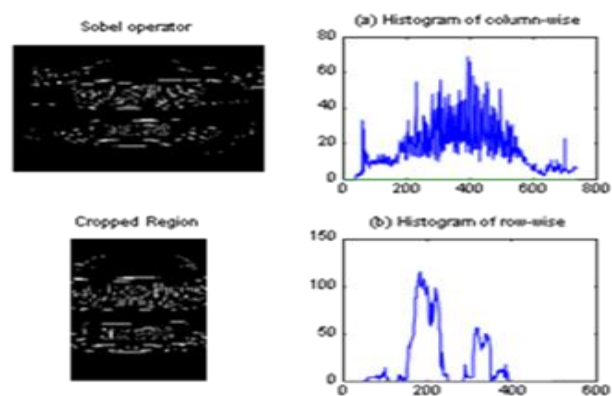Figure 2. Conversion from RGB into grayscale image

Figure 3. License plate localization using Sobel operator edge detection (a) Histogram of column-wise (b) Histogram of row-wise

In the column-wise graph, the histogram calculates the number of pixels at the column part of the diagram. In this histogram, the license plate and character edges are traced. A threshold is set to ignore those that are not license plate candidates. In the row-wise graph, the higher the number of edges in horizontal of the image will create high peaks. With the mean threshold of edges, the location of the license plate can be determined. The final outcome of the license plate localization as shown in Figure 4.

2.3.2. Character isolation

When the location of the license plate is determined, the height and width are calculated and cropped out. The license plate image is cropped directly from the original input image which is in RGB format. After that, the license plate is converted into the grayscale format and finally to binary format for morphological operation. Morphology operation can be used to remove the unwanted noise and isolate the characters from the license plate. In the next step, a rectangle structuring element is created. The dilation and erosion processes are used to separate the foreground and background pixel of the license plate image. The dilation process enlarges the foreground pixel while the erosion shrinks the foreground pixel. By applying subtraction of these two output processes, the edge of the foreground pixel can be obtained. Furthermore, through convolution and contrast color adjusting, the foreground object will be more significant. Finally, the character of the license plate character and background area can be differentiated and filtered out. The final outcome in process character isolation is shown in Figure 5.

Figure 4. Output of license plate localization

Figure 5. License plate isolation using Morphological operation

2.3.3. Character segmentation

After implementing the previous step, the remaining items in the image are only characters. By this, Connected Component Analysis is used to segment the characters using connected pixel in the image. Each of the characters segmented into an individual image for recognition. The output of the character segmentation process is shown in Figure 6.

2.3.4. Character resize and padding

The segmented characters are being resized into 18×8 pixel size to reduce the complexity of the image. Besides that, the image is further padding by 2 pixels on the surrounding of the image to become 22×12 pixel image. Padding is carried out to ensure that all the features are available during the recognition process. The image after padding is shown in Figure 7.
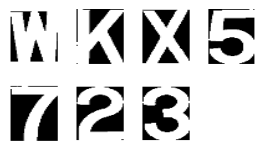

Figure 6. Each of the character is segmented into one image

Figure 7. Characters image is resized and padded into 22×12 pixel image

2.3.5. Normalization

In CNN recognition system, the image is trained using numeric data of the image. The accepted range of the numeric data is from -1 to 1. The min-max normalization is applied to the input images to rescale within the mentioned range. The equation of the min-max normalization is shown below:

2.4. Character recognition

Character recognition is the second stage after preprocessing. In this stage, all the characters are in numeric data form and can be the input for CNN system.

The final output of the preprocessing stage is a set of 22×12 pixels of the image to feed the CNN architecture. The CNN model used in this work is a four-layered architecture. The first and second layer

applied to the fusion method proposed by Mamalet and Garcia. In order to find the best architecture model and the best parameters, 10-fold cross-validation technique has been used.

A total of 80% from the overall sample have been used for 10-fold cross-validation technique. The initial weight used for training and validation are the same for all experiment to ensure a fair evaluation. There are few parameters tested using 10-fold cross validation. The parameters include the number of feature map at each layer, the pattern of connection of the first layer and second layer, type of weight, value of the regulated parameter and the γ-constant (Figure 8).
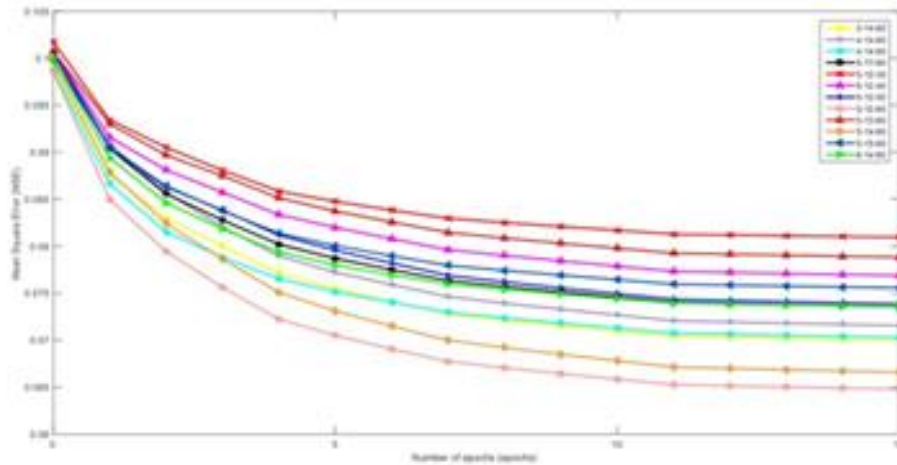

Figure 8. Mean square error versus feature map graph

2.5. LPR system developed in MATLAB GUI

In order to create a user friendly LPR system, the program is developed in MATLAB software and using a MATLAB Graphical User Interface (GUI) for the system representation. As shown in Figure 9, the GUI is a platform to provide a user friendly interface and improve the sustainability of this system.

Figure 9. LPR system GUI

3. RESULTS AND DISCUSSION
3.1. CNN architecture analysis
3.1.1. Feature map

From the Figure 10, there are 12 feature map configurations are being tested through 10-fold cross-validation. The 12 configurations include 3-14-60, 4-14-60, 4-15-60, 5-11-60, 5-12-30, 5-12-40, 5-12-50, 5-12-60, 5-13-60, 5-14-60, 5-15-60 and 6-14-60. Equation (1) and Equation (2) indicates the sizes of the output map (M_x, M_y), in x and y direction and the convolution/subsampling process respectively.

$$M_x^{(l)} = \frac{M_x^{(l-1)} - (K_x^{(l)} - S_x^{(l)})}{S_x^{(l)}}; \ M_y^{(l)} = \frac{M_y^{(l-1)} - (K_y^{(l)} - S_y^{(l)})}{S_y^{(l)}} \qquad (1)$$

$$y_{pj}^{(l)} = f\left(\sum_{i \in M_j^{(l-1)}} \sum_{(u,v) \in K^{(l)}} w_{ji(u,v)}^{(l)} \, 0x_{pi}^{(l-1)}(c + u, r + v) + b_j^{(l)} \right) \qquad (2)$$

where $K = \{(u,v) \in N^2 \mid 0 \le u < k_x; 0 \le v < k_y\}$, $\cdot$ k_x and k_y are the width and the height of the convolution kernels $w_{ji}^{(l)}$ of layer (l) and $b_j^{(l)}$ is the bias of feature map n in layer(l), c and r refers to the current pixel and p refers to the particular training sample. The set $M_j^{(l-1)}$ contains the feature maps in the preceding layer(l-1) that are connected to feature map n in layer (l). The notation f is the activation function of layer(l). The variable u and v describes the horizontal and vertical step size of the convolution kernel in layer(l).
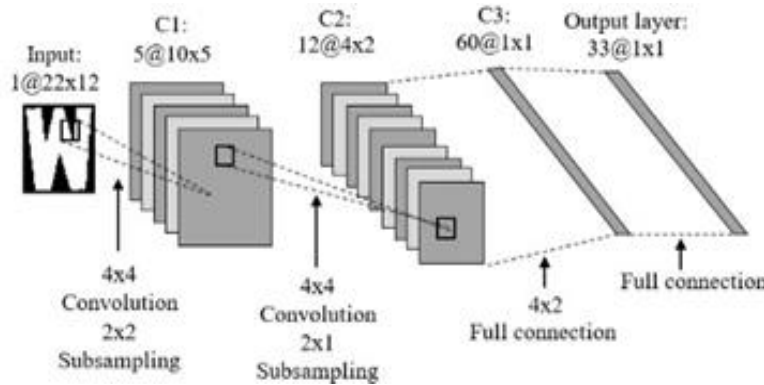

Figure 10. Final architecture of CNN model

The initial feature map in C1, C2 and C3 layers is 5, 14 and 60 features maps. Meanwhile the output layer contains 33 neurons since there are a total of 33 character classes for Malaysian license plate. But the number of feature maps at C1, C2 and C3 layer can vary to suit to the complexity of the input image. Figure 10 is the final result after testing each feature map configuration.

3.1.2. Connection pattern

There are 6 types of pattern connection between C1 and C2 layer as shown in Table 1. The selection of connection in every layer is evaluated by the lowest validation error in these 6 types of connection. Each type of connection is be tested from the first column to the last column in a sequential manner. The lowest validation error among six types of connection will be chosen. Throughout the 10-fold cross-validation, the best connection pattern is as shown in Table 2.

Table 1. Connection between C1 and C2 layers

Layer	Connection				
1st layer	0	0	×	×	×
2nd layer	×	0	×	0	×
3rd layer	0	×	×	×	×
4th layer	×	0	×	0	×
5th layer	×	×	×	×	0
6th layer	0	×	×	×	×
7th layer	0	0	×	×	×
8th layer	0	0	×	×	×
9th layer	0	×	×	×	×
10th layer	0	0	×	×	×
11th layer	0	×	×	×	×
12th layer	×	×	×	×	×

Table 2. Six Type of Connection

Type of connection	Pattern				
Connection 1	×	×	×	0	0
Connection 2	0	×	×	×	0
Connection 3	0	0	×	×	×
Connection 4	×	0	×	0	×
Connection 5	×	×	×	×	0
Connection 6	0	×	×	×	×

3.1.3. Type of weight selection

The weight initialization is carried out by using 10-fold cross-validation. There are 4 types of weight being tested namely Gaussian, Nguyen, Fan-in and uniform. From Table 3, the Gaussian weight produces the best accuracy which is 81.48% for recognition. Therefore, Gaussian weight has been used for the rest of the testing.

Table 3. Type of weight versus accuracy

Types of weight	Accuracy (%)
Gaussian	81.48
Nguyen	48.00
Fan-in	0.00
Uniform	59.00

The learning algorithm used in training the CNN is an enhanced version of Stochastic Diagonal Levenberg Marquadt (SDLM) [35] algorithm. This learning algorithm is better than standard backpropagation in which it avoids the gradient from getting trap into the local minima. As a result when SDLM algorithm is used, a smoother error gradient is achieved. In the mentioned algorithm, there are two parameters need to be tuned namely regularization, parameter µ and the γ parameter.

3.1.4. Regularization parameter

The regularization parameter is the parameter of the learning algorithm which is SDLM. This parameter can vary from 0.04 until 0.09. The outcome of each value of the regularization parameter is shown in Figure 11.

From Figure 11, the regularization parameter = 0.04 has produce the lowest mean square error (MSE) in comparison to others. Therefore, 0.04 is the best value to be used for license plate recognition. According to the graph, the feature map 5-12-60 performance is higher than other configuration which also including the initial feature 5-14-60. Through this experiment, a best feature map configuration is obtained which is 5-12-60.
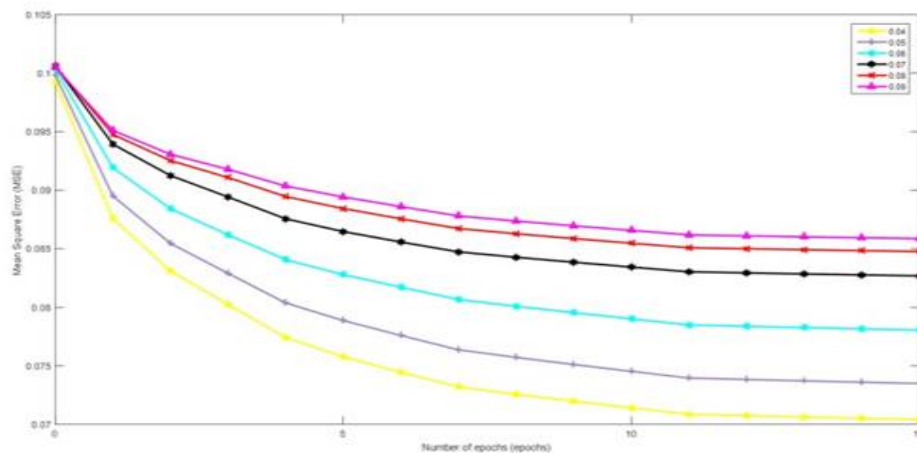

Figure 11. Mean square error versus regulation parameter graph

3.1.5. y–constant

γ-constant is also the parameter from SDLM learning algorithm. The value of the γ-constant can either be 0.1, 0.01 or 0.001. The result of the performance is obtained by the accuracy of recognition during validation by using 10-fold cross-validation experiment. The Table 4 showed that the γ-constant 0.01 having the highest accuracy of recognition which is 81.48%. So that, γ-constant = 0.01 is used in the CNN recognition system.

This result can be further improved if the problem at the preprocessing part has solved. Table 5 shows the accuracy of preprocessing and CNN recognition.

Table 4. γ-constant versus Accuracy	
Value	Accuracy (%)
0.1	70.37
0.01	81.48
0.001	77.78

Table 5. Accuracy of LPR System	
Process	Accuracy (%)
Preprocessing	74.7% out of 300 samples
CNN recognition	94.6% out of 528 samples

3.1.6. Performance of the designed LPR system

In this paper, two types of accuracies are taken at different stages. The first accuracy is taken at the preprocessing part the system and the second one is taken at the classification stage. The classification result is taken according to the number of characters successfully recognized.

Table 5 describes that the preprocessing part achieves 74.7% out of 300 samples tested which does not achieve the expectation level. The preprocessing algorithm needs to be further improved in order to effectively filter out noises such as environment factor (illumination) to achieve higher accuracy. The preprocessing process is prominent because whenever the LPR system failed to preprocess the input image correctly, the accuracy will be affected at the classification stage.

CNN able to achieve 94.6% out of 528 samples tested. This CNN recognition designed model has achieved the expectation level that can practically use in future. The CNN recognition can be considered as a robust recognition technique to be used in license plate character recognition.

4.   CONCLUSION

A LPR system using the CNN recognition method is successfully developed. The preprocessing stage of LPR system that is developed in MATLAB software has successfully merged with the CNN recognition system in C language. This system needs improvement at the preprocessing stage to achieve a better accuracy level. For future recommendation, this proposed work can be used to enhance the AES in Malaysia. Currently the AES is used to capture the vehicle that exceeds the speed limit only. The analysis of the captured images is done by human.

ACKNOWLEDGEMENT

REFERENCES

[1]   N. A. Bakar, *et al.*, "Malaysian Vehicle License Plate Recognition Using Double Edge Detection," *2012 IEEE International Conference on Control System, Computing and Engineering, Penang*, pp. 422-426, 2012.
[2]   M. F. Zakaria and S. A. Suandi, "Malaysian Car Number Plate Detection System Based on Template Matching and Colour Information," *International Journal on Computer Science and Engineering*, vol/issue: 2(4), pp. 1159–1164, 2010.
[3]   W. W. Keong and V. Iranmanesh, "Malaysian Automatic Number Plate Recognition System using Pearson Correlation," *2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Batu Feringghi*, pp. 40–45, 2016.
[4]   S. R. Soomro, *et al.*, "Vehicle Number Recognition System for Automatic Toll Tax Collection," *2012 Int. Conf. Robot. Artif. Intell. (ICRAI 2012)*, pp. 125–129, 2012.
[5]   F. H. C. Tivive and A. Bouzerdoum, "A Face Detection System Using Shunting Inhibitory Convolutional Neural Networks," *Neural Networks, 2004. Proceedings. 2004 IEEE Int. Jt. Conf.*, vol. 4, pp. 2571–2575, 2004.
[6]   C. Garcia and M. Delakis, "Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol/issue: 26(11), pp. 1408–1423, 2004.
[7]   M. Delakis and C. Garcia, "Training Convolutional Filters for Robust Face Detection," *Neural Networks Signal Process. - Proc. IEEE Work.*, vol. 2003, pp. 739–748, 2003.
[8]   N. Farrugia, *et. al.*, "Fast and Robust Face Detection on a Parallel Optimized Architecture Implemented on FPGA," *IEEE Trans. Circuits Syst. Video Technol.*, vol/issue: 19(4), pp. 597–602, 2009.
[9]   C. Poulet, *et al.*, "CNP: An FGPA-based Processor for Convolutional Networks," *FPL 09 19th Int. Conf. F. Program. Log. Appl.*, vol/issue: 1(1), pp. 32–37, 2009.
[10] N. Farrugia, *et al.*, "Design of a Real-Time Face Detection Parallel Architecture Using High-Level Synthesis," *EURASIP J. Embed. Syst.*, vol. 2008, pp. 1-9, 2008.
[11] P. Buyssens and M. Revenu, "Learning sparse face features: Application to Face Verification," *Proc. - Int. Conf. Pattern Recognit.*, pp. 670–673, 2010.
[12] S. Chopra, *et al.*, "Learning a Similiarty Metric Discriminatively, With Application to Face Verification," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 349–356, 2005.

[13] H. Ghiassirad and M. Teshnehlab, "Similarity Measurement in Convolutional Space," *IS'2012 - 2012 6th IEEE Int. Conf. Intell. Syst. Proc.*, pp. 250–255, 2012.

[14] Y. N. Chen, *et al.*, "The Application of a Convolution Neural Network on Face and License Plate Detection," *Proc. - Int. Conf. Pattern Recognit.*, vol. 3, pp. 552–555, 2006.

[15] G. B. Huang, *et al.*, "Learning Hierarchical Representations for Face Verification with Convolutional Deep Belief Networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2518–2525, 2012.

[16] S. S. Liew, *et al.*, "Gender Classification: A Convolutional Neural Network Approach," pp. 1248–1264, 2016.

[17] F. Hing, *et al.*, "A Gender Recognition System using Shunting Inhibitory Convolutional Neural Networks," *Int. Jt. Conf. Neural Networks (IJCNN '06)*, pp. 5336–5341, 2006.

[18] F. H. C. Tivive and A. Bouzerdoum, "A Shunting Inhibitory Convolutional Neural Network for Gender Classification," *Proc. - Int. Conf. Pattern Recognit.*, vol. 4, pp. 421–424, 2006.

[19] S. F. Abdullah, *et al.*, "Multilayer Perceptron Neural Network in Classifying Gender using Fingerprint Global Level Features," vol/issue: 9(9), 2016.

[20] C. H. Teo, *et al.*, "A Novel Approach to Improve the Training Time of Convolutional Networks for Object Recognition." Retrive from http://users.cecs.anu.edu.au/~chteo/pub/TeoTayLai05.pdf

[21] F. J. Huang and Y. LeCun, "Large-scale Learning with SVM and Convolutional Nets for Generic Object Categorization," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, pp. 284–291, 2006.

[22] M. M. Piramli, *et al.*, "Rice Grain Grading Classification Based On Perimeter Using Moore-Neighbor Tracing Method," vol/issue: 8(2), pp. 23–27, 1843.

[23] S. S. Ahranjany, *et al.*, "A Very High Accuracy Handwritten Character Recognition System for Farsi/Arabic Digits Using Convolutional Neural Networks," *Proc. 2010 IEEE 5th Int. Conf. Bio-Inspired Comput. Theor. Appl. BIC-TA 2010*, pp. 1585–1592, 2010.

[24] S. Arora, *et al.*, "Performance Comparison of SVM and ANN for Handwritten Devnagari Character Recognition," *International Journal of Computer Science Issues*, vol/issue: 7(3), pp. 1–10, 2010.

[25] H. Swethalakshmi, *et al.*, "Online Handwritten Character Recognition of Devanagari and Telugu Characters using Support Vector Machines," *Guy Lorette. Tenth International Workshop on Frontiers in Handwriting Recognition, La Baule (France), Suvisoft*, pp. 1–6, 2006.

[26] F. H. C. Tivive and A. Bouzerdoum, "Texture Classification using Convolutional Neural Networks," *TENCON 2006 - 2006 IEEE Reg. 10 Conf., Hong Kong*, pp. 14–17, 2006.

[27] A. R. Syafeeza, *et al.*, "A Review of Finger-Vein Biometrics Identification Approaches," *Indian Journal of Science and Technology*, vol/issue: 9(32), 2016.

[28] D. R. Tobergte and S. Curtis, "A Unified Architecture for the Detection and Classification of License Plates," *J. Chem. Inf. Model.*, vol/issue: 53(9), pp. 1689–1699, 2013.

[29] S. A. Radzi and M. Khalil-hani, "Character Recognition of License Plate Number Using Convolutional Neural Network," *VIC'11 Proceedings of the Second international conference on Visual informatics: sustaining research and innovations, Selangor, Malaysia*, pp. 45–55, 2011.

[30] Z. Zhao, *et al.*, "Chinese License Plate Recognition Using a Convolutional Neural Network," *2008 IEEE Pacific-Asia Work. Comput. Intell. Ind. Appl.*, pp. 27–30, 2008.

[31] Y. Wen, *et al.*, "An Algorithm for License Plate Recognition Applied to Intelligent Transportation System," *IEEE Trans. Intell. Transp. Syst.*, vol/issue: 12(3), pp. 830–845, 2011.

[32] X. Zhai, *et al.*, "License Plate Localisation based on Morphological Operations," *11th Int. Conf. Control. Autom. Robot. Vision, ICARCV 2010*, pp. 1128–1132, 2010.

[33] S. Chakraborty, "An Improved Template Matching Algorithm for Car License Plate Recognition," *International Journal of Computer Applications*, vol/issue: 118(25), pp. 16–22, 2015.

[34] Y. Ma, *et al.*, "A new algorithm for characters segmentation of license plate based on variance projection and mean filter," *Proc. 2011 IEEE 5th Int. Conf. Cybern. Intell. Syst. CIS 2011*, pp. 132–135, 2011.

[35] A. R. Syafeeza, *et al.*, "Convolutional Neural Networks with Fused Layers Applied to Face Recognition," *International Journal of Computational Intelligence and Applications*, vol/issue: 14(3), 2015.

BIOGRAPHIES OF AUTHORS

Muhamad Marzuki Piramli was born in Malaysia in year 1992. He was completed bachelor degree in Computer Science (Artificial Intelligent) in year 2015 from Technical University of Malaysia Malacca (UTeM). Present he is pursuing MSc. in Electronic and Computer Engineering at Technical University of Malaysia Malacca (UTeM). His Research includes Deep Learning, Pattern Recognition, Machine Learning and Artificial Intelligent.

Dr. Syafeeza Ahmad Radzi was born in Malaysia in year 1981. She was awarded degree and M. Eng. in Electrical and Electronics Engineering in Technology University of Malaysia (UTM) Johor Bahru, Malaysia. She completed Ph. D degree in Electrical Engineering from Technology University of Malaysia (UTM) in year 2014. Currently, she is working as Senior Lecturer in the Department of Computer Engineering at Technical University of Malaysia Malacca (UTeM). Her research area in Embedded System, Pattern Recognition, Machine Learning, Image Processing.