License Plate Recognition using Multilayer Neural Networks

Siti Norul Huda Sheikh Abdullah
(mimi@sun1.ftsm.ukm.my)

Abstract

Vehicle license plat recognition has been a much studied research area in many countries. Due to the different types of license plates being used, the requirement of an automatic license plate recognition system is rather different for each country. In this paper, an automatic license plate recognition system is proposed for Malaysian vehicles with standard license plates based on image processing, feature extraction and neural networks. The image processing library is developed in-house which we referred to as Vision System Development Platform (VSDP). The Kirsch Edge feature extraction technique is used to extract features from the license plates characters which are then used as inputs to the neural network classifier. The neural network model is the standard multilayered perceptron trained using the back-propagation algorithm. The prototyped system has an accuracy of about 91%, however, suggestions to further improve the system are discussed in this paper based on the analysis of the error.

Keywords: License plate recognition, feature extraction, classification.

1. Introduction

Automatic license plate recognition system is an important area of research due to its many applications. For local authorities license plate recognition is required for the purposes of enforcement, border protection, vehicle thefts, automatic toll collection, and perhaps traffic control. For others, automatic license plate recognition system can be applied to access control in housing areas, automatic parking control and marketing tools in large shopping complexes, and perhaps for surveillance.

Among the commercial license plate recognition systems available worldwide are Car Plate Recognition by J.A.G. Nijhuis et. al. [1], Car Plate Reader (CPR) by Rafael et. al. [2], Optical Car Recognition by Emiris and Koulouriotis [3] and Automatic Number Plate Recognition(ANPR) by Shyang-Lih Chang et. al. [4] and Mehmwet Sabih Aksoy et. al.[5].

In Malaysia, vehicles license plates are in the form of single or double line with normal fonts which comprise of perhaps 95% of the all the vehicles. There are also special fonts as depicted below (Fig 1.0 b).

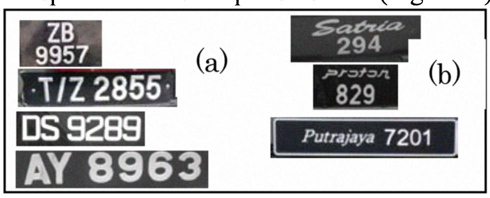

Fig. 1.0 (a):Samples of common Malaysia license plates (b) Samples of special Malaysia license plates.

Signal processing is considered as a brain of recognition system [6] while License Plate Recognition (LPR) is a subset of signal or vision processing technology or an Intelligent Transport System (ITS), used to identify vehicles by their plate numbers that has been widely used in leading industrial countries as well as in developing countries like Malaysia. LPR system, an Optical Character Recognition (OCR) for Cars[7], normally integrates various input and output devices like access control system and camera or CCTV to capture images of vehicle plate numbers. An example of an entry scenario at car park lane, one need to insert parking card onto card reader but at the same time a CCTV will record time and its plate number for security purposes, and therefore the gate will then be opened. Most of the LPR can be applied for security and traffic application such as parking, access control, border control, traffic control, tolling, marketing tool, enforcement and travel. However, practically the LPR system differs from each country due to several issues such as format and colour of license plate.

LPR normally consists of a camera, illumination, frame grabber, computer, recognition software, hardware (input output adapters) and database as illustrated in Fig 1.1. LPR employs real time plus artificial intelligence algorithm like hybrid system or Neural Network (NN) which recognizes significant plate numbers and records in the refined databases. This dedicated LPR software covers at least five major processes consecutively; Capturing, Pre-Processing, Segmentation, Feature Extraction and Classification as shown in Fig 2. Usually, targeted functions and specifications that will be embedded into the LPR system are fast recognition alphabet and number with high accuracy recognizing both front and back side, 24 hours non-stop operation and alarm message send out after recognition.

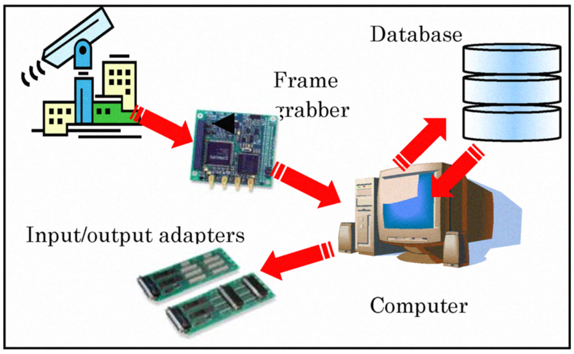

Fig 1.1: Elements in LPR

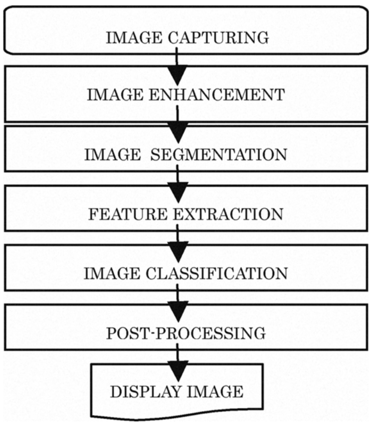

Fig 2: Process in image processing

2. Image Segmentation

Image segmentation is a process that separates words to single characters for easy identification [8]. In this project, segmentation involves a process of separating a collection of character that has been filtered; to a sequence of characters that will be used in the feature extraction stage. This step is very significant due to overlapping characters that form the license plate. There are three main forms of characters that are overlapping vertically, ligature, diacritics, horizontal overlap, and two connected characters. The task will be more difficult for those different forms of which are joined.

At the moment, VisionPlate II applies clustering technique to identify important blobs. After processing image using simple image enhancement technique like Fixed filter, Minimum Filter, Opening and dependent threshold for the VisionPlate II image enhancement which are provided in VSDP library (Vision System Development Platform). VSDP is a library that has been developed by CAIRO, UTMKL researchers.

After applying above image enhancement, the image is segmented using horizontal scan line profiles and clustering technique. Thoroughly each image is transformed into blob objects and its important information such as location, height and width, are being analyzed by the VisionPlate II for the purpose of cluster exercising and choosing the best cluster with winner blobs. The blobs are clustered when difference between blob and cluster heights and difference between maximum Y value of the cluster and blob are less than a constant time to cluster's height as stated in clustering algorithm. Please refer to the clustering algorithm in section 2.1 and picture depicted in Fig 3.1 and 3.2. Then these winner blobs are extracted its feature individually before permitting to recognition or classification phase.

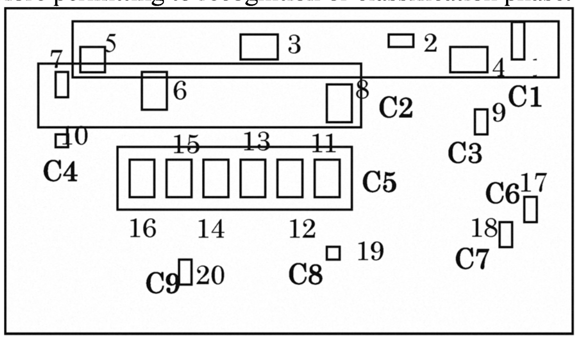

Fig 3.1: Image Segmentation using clustering approach.

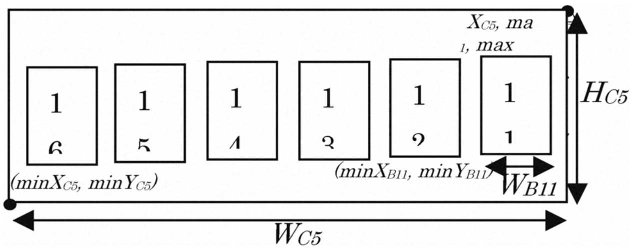

Fig 3.2: Important information for clustering approach.

2.1 Clustering algorithm.

Input: Set original image into a buffer, B_1.

Output : Get winner clusters and blobs, C_n, Bb_{1-m}

Step 1: Calculate total of blobs in the image, n.

Step 1.1: From 0 until n, then keep information like $minx_n$, $minY_n$, $maxX_n$, $maxY_n$, $height_n$, $width_n$ for each blobs into an array.

Step 2: Cluster each blobs when difference between blob height, $height_{Bm}$ and cluster height, H_{cn} and difference between maximum Y value in cluster, $maxY_{cn}$ and maximum Y value of blobs, $maxY_{bm}$ are less than a constant time to the cluster height, H_{cn}.

$$|maxY_{Bi} - maxY_{Ci}| < (\alpha \times H_{Ci}) \quad (1)$$

$$|H_{Ci} - H_{Bi}| < (\alpha \times H_{Ci}) \quad (2)$$

where α value is 0.3, 0.5, 1 or 2.

Step 3: Choose the cluster, C_n which has the maximum size of blobs, $C_{size\ m}$

Step 3.1: Check distance between each winner blobs, $maxX_n$ and $minx_{n+1}$.

Step 3.2: Sort the winner blobs according to its $minX_n$.

1-4244-0220-4/06/$20.00©2006 IEEE

Step 3.3: Segment all sorted winner blobs individually.

Step 4: Finish.

3. Feature Extraction

Feature extraction is described as functions of the measurements performed on a class of objects that enable class to be distinguished from other classes in the same general category. One of feature extraction objective is to grab only essential and distinguished information or characteristics of the each character to be easily recognized later [8]. Some researchers applied thinning or skeleton[3], Laplacian Edge detector[5], Minimum Area [3], Prewits, Robinson and Sobels [10]edge detector. In our research we concertrated on Kirsch Edge Detection.

3.1 Kirsch Edge Detection. Basically kirsch edge detector have eight different kernels to detect eight different directions of edges. For example, consider the matrix below (Fig 4), on the right is left vertical, left vertical, right vertical, bottom horizontal, top horizontal, bottom left diagonal, top left diagonal, bottom right diagonal and top right diagonal. Kirsch Edge Detection is a simple algorithm for first-order differential edge detection. This edge detector is used to detect four directional edges more accurately than other detectors such as Prewitt and Sobel which considers all the eight-neighborhood pixels. The non-linear edge enhancement algorithm defined by Kirsch is shown as follows:

(3a)

(4)

The $G(i,j)$ is the gradient of the pixel (i,j). The subscripts of A are represented as the neighborhood pixels for the (i,j) as shown in Fig 5.We can calculate the directional feature vectors for horizontal (H), vertical (V), right-diagonal(R) and left-diagonal (L) directions as follows:

(4.1a)
(4.1b)
(4.1c)
(4.1d)

(a)

$$\frac{1}{15}\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \end{bmatrix} \frac{1}{15}\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \end{bmatrix}$$

(b)

$$\frac{1}{15}\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \end{bmatrix} \frac{1}{15}\begin{bmatrix} 5 & 5 \\ -3 & 0 \end{bmatrix}$$

(c)

$$\frac{1}{15}\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \end{bmatrix} \frac{1}{15}\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \end{bmatrix}$$

(d)

$$\frac{1}{15}\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \end{bmatrix} \frac{1}{15}\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \end{bmatrix}$$

Fig.4: Matrix mask for (a)left and right vertical (b) bottom and top horizontal(c)bottom and top left diagonal (d)bottom and top right diagonal.

A_0	A_1	A_2
A_7	(i,j)	A_3

Fig 5: Example showing the eight neighbors of pixel (i,j).

The equations above can be replaced by simple convolution masks operation as given in Fig. 5 and the scale factor of 1/15 was suggested by Pratt [11]. The edges extracted from different classes of characters are not the same and the operation speed is also acceptable. Thus, it could be used as the feature extractor for the character recognition. The results of the Kirsch detectors are shown in Fig. 6 and 7. Note that after the Kirsch edge detection, the image will change from binary to gray level scale.

Fig 6: Original Image for character '4', '6', 'E' and 'G' before cropping each object.

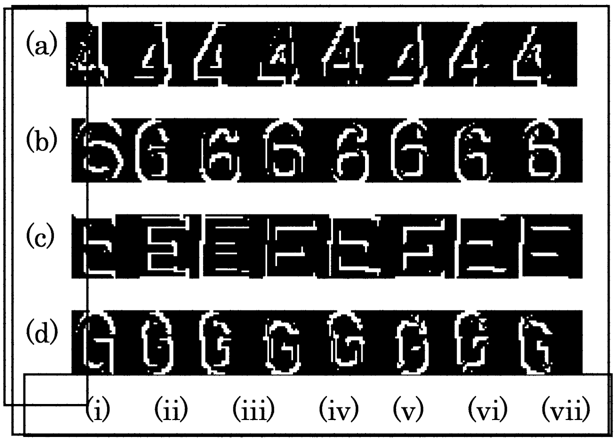

Fig 7: Feature extracted images using Kirsch Edge Detection for alphanumeric (a) 4, (b) 6, (c) G and (d) while its' 8 types are (i) left vertical (ii) right vertical (iii)bottom horizontal (iv) top horizontal (v) bottom left diagonal (vi) top left diagonal (vii) bottom right diagonal (viii) top right diagonal.

3.1.1 Kirsch Edge algorithm

Input: Set Original Image into a buffer, *B1*.

Output : Display and binarized feature extracted image.

Step 1: For counter, $i = \{1,2,3,4,5,6,7,8\}$

Step 2: Clone original image *B1* into new buffer, B2.

Step 3: Create new Kirsch Edge kernel with size (3,3) for each eight directional edges K_j where $j = \{1,2,3,4,5,6,7,8\}$.

Step 3.1: Set kernel for left vertical (K_1), right vertical(K_2), bottom horizontal (K_3), top horizontal (K_4), bottom left diagonal (K_5), top left diagonal (K_6), bottom right diagonal (K_7) and top right diagonal (K_8).

Step 4: Do convolution buffer *B2* with chosen kernel directions of kernel, K_j.

Step 5: Display and binaries each feature extracted image.

Step 6: Finish.

A minor research has been conducted to select the best kernel in Kirsch Edge Detector and we found that right vertical, top horizontal, top left diagonal and top right diagonal are the best features to represent character images and inputs to neural network.

3.Image Classifications

We apply neural network (Fig. 8) to classify these images and recognize the characters. Here we explain briefly foundation of NN. Mui et al. [12] presented two important characteristics in NN: learning and generalization. Learning process associates with network architecture that will change the connection structure between units and signal strength in the connection structure. Hagan et al. [12] proposes a multi-input processing model as depicted in Fig 4.

Every input, x_1, x_2,…, x_n has its respective weight $w_{1,1}$, $w_{1,2}$, …, $w_{1,n}$ from weight matrix w. This neuron has bias b that will be accumulated with clean input to produce total neuron input value.

Total neuron input value is used in the activation function f, and produces one scaled output neuron, a that can be represented by:

$$a = f\left(\sum_{i=1}^{n} w_{1,i} x_i + b\right) \quad (3)$$

Output a value depends on the activation function used. Basically there are two types of activation functions: linear and non-linear. Activation function either Binary sigmoid, Bi-polar sigmoid or Hyperbolic tangent, which is suitable with the type of problem solving and desired output range, shall be applied onto the network [16].In our case we applied Binary sigmoid. We also used random weight control for the first network initializing even though there are other types like Nguyen Widrow [17] and Genetic algorithm [15].

After a few experiments conducted (Table 1), we found that using five features: original image and kirsch edge kernel 2, 4, 6 and 8 with 10x10 image size is the most essential input numbers for the neural network scheme. Meanwhile, 200 are the most optimum hidden nodes. We have trained on 200 image sets and stopped training when its mean square errors have reached to 0.0026 values. Since we are dealing with Malaysian license plate, the output nodes have been increased from 33 up to 36 which covers all roman alphabets (except O) , numbers (from 0 to 9) and backlash ("/ ") .

Table 1: Neural Network Scheme.

Input nodes	5 types x (10 x 10 pixel)	500
Hidden nodes		200
Output	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,G,H,I,J,K,L,M,N,P,Q,R,S,T,U,V,W,X,Y,Z and / .	36
Learning rate		0.05
Minimum Error rate		0.0026

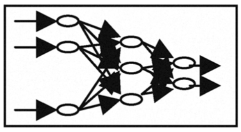

Fig 8. NN Multi-input processing

4. Discussions

There were two experiments conducted; fixed (value for threshold is 130) and different threshold value experiments. Each experiment had been run onto 1000 off-line image data. Here, we only stated classification, feature extraction and classification time because the experiment has been carried out automatically. These images are captured from frontal and back of Malaysia cars. This time we only concentrated on Malaysia standard car plate images which taken surrounding Kuala Lumpur, Selangor, Pahang, Terengganu and Perak. VisionPlate II is developed using Microsoft Visual C++ and VSDP library. VSDP library is a library for image processing and it has consistently developed and updated by CAIRO.

As depicted in Table 2, Classification has consumed the highest time which is 2247.68 ms while feature extraction falls the second with 472.02 ms and segmentation is the least with 0.5ms. Classification gains most time due to neural network processing time which requires connecting to the weight database and calculating the current image's weight.

Table 2: Average time for five separated of fixed threshold experiment.

Sample size		200	200	200	200	200	1000	Rate
Type	SET	1	2	3	4	5	Total	
Time Ave-rage:	Segmentation	3.17	2.39	9.95	5.23	5.02	25.76	0.5
	Feature Extraction	412.88	413.54	552.64	456.6	524.46	2360.12	472.02
	Classification	2192.05	2115.74	2338.44	2288.83	2303.34	11238.4	2247.68

From Table 3, out of a thousand images that were been analyzed, 803 images have perfectly recognized for fixed threshold experiment. This result increased to 919 when different threshold values were used. Therefore, both experiments accuracy percentage are 80.3% and 91.9% correspondingly.

Table 3: Accuracy percentage for fixed and different threshold experiment**.**

	threshold		average
	fixed (130)	different	
total sample data	1000	1000	1000
no of correct	803	919	861
correct percentage	80.3 %	91.9 %	86.1%
total segmentation error	121	10	65.5
Segmentation error percentage	61.4%	12.34 %	36.88 %
classification error	76	71	73.5
Classification error percentage	38.58%	87.65 %	63.12 %

Even though, VisionPlate II accuracy percentage has achieved more than 80%, they are several issues to be tackled in the case error analysis such as segmentation and classification issues. From table 3, segmentation error percentage for fixed threshold is about 61.4% while classification error rate is 38.58%. However, when different threshold values are used, its segmentation error percentage has reduced significantly to 12.34% and caused the classification error percentage increased to 87.65%. Here, we can assume that by applying different threshold or perhaps adaptive threshold values can reduce segmentation error percentage. Furthermore, VisionPlate II also needs to give attention to classification errors because adaptive threshold did not show any significant improvement.

Segmentation errors are categorized into five classes: NotFound, Miss_1, Miss_2, Miss_>2 and Extra. Meanwhile classification errors are divided into four categories: Wrong_1, Wrong_2, Wrong_>2 and Wrong_seq. Description of each errors are explained briefly in Table 4. Samples of interfaces of those errors are also depicted in Fig. 10.1 and 10.2.

Referring to the segmentation problems of Table 4, there were 26 errors for Type Miss_1, 30 errors for Type Miss_2, 57 errors for Type Miss_>2 and 8 errors for Type Extra in fixed threshold value experiment. These errors were occurred may due to restrictions in clustering approach. Inappropriate threshold causes two and more characters connected and width of the blobs is greater than the height of the blobs. As a result, these connected character blobs will not consider as winner blobs and become missing. Clustering success is closely related to the constant value that has been set for grouping the blobs (Refer to formula 1 and 2). If the constant value increases, VisionPlate II surprisingly can detect almost more than 20 but less that 50 degree of skewed license plate. However, the drawback is sometimes unnecessary blobs will also consider as winner blobs and this error falls into category Extra. On the other hand, if the constant value reduces, this may also lead to missing blobs like Miss_1, Miss_2 and Miss_>2 errors.

Fortunately, these segmentation errors were reduced dramatically when different threshold values were used. From table 4, you can see that out of 121 errors occurred in fixed threshold experiment, only 10 remained as errors when different threshold values were adapted. Therefore, an adaptive threshold system is highly required to be developed such as Otsu Threshold, NN Threshold or Rule-based Threshold.

Table 4: Type of errors for fixed and different threshold experiment.

Error		threshold		average
Type	Description	fixed	different	
NotFound	Cannot find license plate	0	1	0.5
Miss_1	Miss 1 character	26	2	14
Miss_2	Miss 2 characters	30	2	16
Miss_>2	Miss more than 2 characters	57	3	30
Extra	More than actual characters	8	2	5
total segmentation errors		121	10	65.5
Wrong_1	Wrong 1 character	56	52	54
Wrong_2	Wrong 2 characters	9	6	7.5
Wrong_>2	Wrong more than 2 characters	11	10	10.5
Wrong_seq	Wrong sequence of characters	0	3	1.5
total classification errors		76	71	7

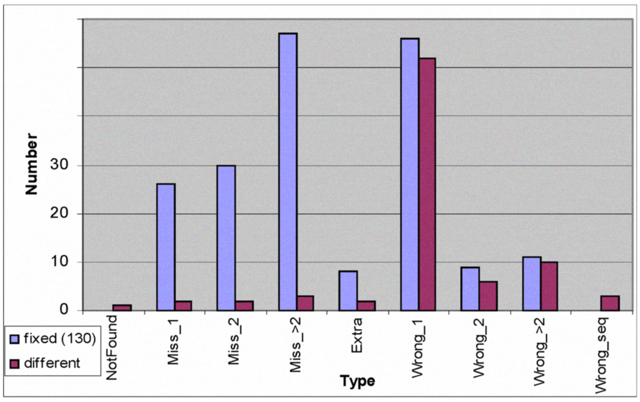

Fig 9: Overall type versus number of errors for fixed and different threshold graph.

On the other hand, classification is another serious issue as depicted in Fig.9 and Table 3 and 4. There were 56, 9, 11 and 0 errors for Type Wrong_1, Wrong_2, Wrong_>2 and Wrong_seq consecutively when fixed threshold experiment was conducted. These errors almost remained the same (except for Type Wrong_seq) even though different threshold values were used. Errors that fall under category classification may due to segmentation techniques. Quite a number that Type Wrong_1, Wrong_2 and Wrong_>2 were wrongly recognized because the license plate images were skewed or rotated. Therefore, some of the letters were misclassified. For example, several characters that look similar were detected vice versa like character B or 3 detects as 8, character 5 detects as 6, character 6 detects as G (as depicted in Fig. 7b and 7d), character A detects as 4, character I detects as 1.

Apart from that, these errors may also due to inappropriate feature extraction technique. Kirsch edge detector is intolerance to rotated images. Kirsch Edge detector also fails to distinguish certain character like 6 and G. As depicted in Fig 7b and 7d, feature representation for 6 and G may return the same binary value. As a result, Kirsch Edge might lead to letter misclassification.

6. Suggestions

Firstly, both fixed and different threshold experiment shows that the performance increases if the appropriate threshold is applied before segmenting the characters. Adaptive threshold like Otsu Threshold or Otsu with a revised formula need to be developed for reducing segmentation errors that may lead to misclassification later.

Secondly, segmentation algorithm should solve geometric issues from the very beginning. Geometric approach can re-correct the position of coordinates and aid to arrange the character in proper order. Furthermore, geometric approach also helps maintaining uniqueness of each letter characteristics by correcting its structure. Besides that, there are three main approaches of segmentation, which are Histogram Profile Projection or HPP, Connected Components Labeling (CCL), and Determining of Segmentation Points (DSP)[18]. HPP can be used to segment text-to-text lines, then to words. CCL can gather all contours of connected components. Meanwhile, DSP is stressed on the determination of definitive segmentation points by searching junction of segments between characters. Combination of these three approaches can form better solution in segmentation phase.

The third problem may cause by feature extraction. Feature extraction has a good correlation with the success of the recognition. Using other feature extraction, which explains and represents better nature of each character is required. Feature extraction is divided into three styles; grayscale image, binary image or vector (skeleton) image[18]. Grayscale image consists of several techniques like template matching, deformable templates, unitary transform, zoning, geometric moments and Zernike moments. For binary feature extraction, techniques above are used similarly to grayscale image and plus contour profiles, spline curve and fourier descriptors. Vector image includes feature extraction technique such as graph descriptors and discrete feature.

Lastly, the recognition using neural network can lead to misclassification if errors in segmentation and feature extraction are not solved independently.

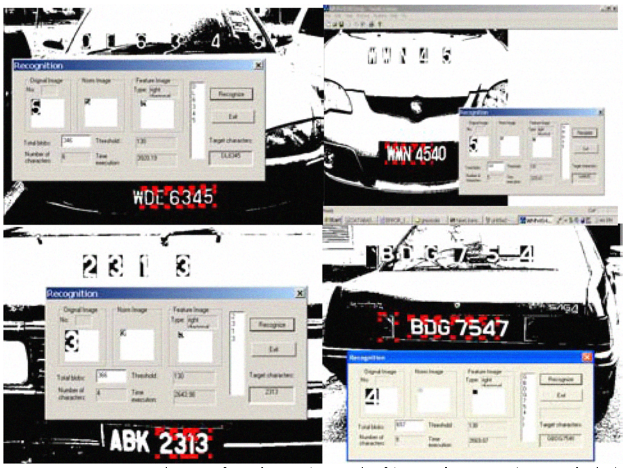

Fig 10.1: Samples of mis_1(top left), miss_2 (top right), miss_>2 (bottom left) and extra character (bottom right).

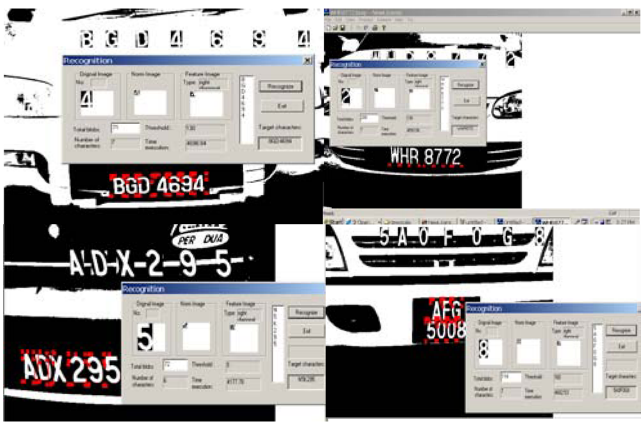

Fig 10.2: Samples of wrong_1(top left) , wrong_2 (top right), wrong_>2 (bottom left) and wrong_seq (bottom right).

Otherwise, perhaps other technique to classify can be used like Trace Transform, Polynomial and Bayesian classification.

5. Conclusions and Acknowledgment

This paper has generally discussed on concept of license plate recognition, segmentation, feature extraction approach and neural network technique. In conclusion, we can conclude that classification has significantly raised more problems compared to segmentation. Major adjustment must be made to reduce recognition errors. These errors may origin to insufficient segmentation algorithm or inefficient feature extraction method (Kirsch Edge Detector).

Special thanks to my friends: Norzi, Fari, Norzu and Dilah who had been hardworking in collecting data in various states.

References

1. J.A.G. Nijhuis, M.H ter Brugge, K.A. Helmholt. (1995). Car License Plate Recognition with neural network and fuzzy logic. IEEE Proceedings of International Conference on Neural Networks. 27 Nov-1 Dec. (5):2232-2236.
2. Rafael, A, Barroso, J., et al.1997. Number Plate Reading Using Computer Vision. Industrial Electronic, 1997 ISIE'97, Proceedings of IEEE International Symposium on Publish, vol.3, 1997, pp 761-766.
3. Emiris, D.M, Koulouriotis, D.E.(2001).Automated optic recognition of alphanumeric content in car license plates is semi-structured environment. Proceedings of International Conference on Image Processing. 7-10 Oct. (3):50-53.
4. Shyang-Lih Chang, Li-shien Chen, Yun-Chung Chung, Sei-Wan Chen (2004). Automatic license plate recognition. Intelligent transportation system. IEEE Transaction. Vol(5):42-53.
5. Mehmwet Sabih Aksoy, Gultekin Cagil, Ahmet Kursat Turker (2000) Number-plate recognition using inductive learning. Robotics and Autonomous systems.(33):149-153. www.elsevier.com/locate/robot. Elsevier science B.V.
6. Grattoni, P., Pettiti, G., Rastello, M.L. (1999). Experimental set-up for the characterization of automated number-plate recognizers. Measurement. (26):103-114. http://www.elsevier.com/locate/measurement.
7. Hofman, Y. (2004) License Plate Recognition - A Tutorial (online) http://www.licenseplaterecognition.com/#Elements (access date: 13 May 2004).
8. B. Al-Badr, S.A.Mahmoud (1995) Survey and bibliography of Arabic optical test recognition. Signal Processing. Vol (41) pg 49-77.
9. Tay, Y.H., Khalid, M (1997) Comparison of Fuzzy ARTMAP and MLP Neural Networks in Hand-written Character Recognition, Pre-prints of

International Federation of Automatic Control (IFAC) Symposium on Artificial Intelligent in Real - Time Control 1997 (AIRTC'97), 1997, pp.363-37.

10. Safraz, M. Ahmed, M.J. Ghazi, S.A. (2003). Saudi Arabian license plate recognition system. Proceedings of International Conference on Geometric Modelling and Graphics. 16-18 July. Pg 34-41.
11. W.K Pratt (1978) Digital Image Processing. New York: Wiley.
12. Mui, L.; Agrarwal, A.; Gupta, A.; & Wang, P.S.P. (1994) An Adaptive Modular Neural Network with Application to Unconstrained Character Recoignition. In Document Image Analysis, Bunke, H.; Wang, P.S.P.; & Baird, H. S. World Scientific 16:1189-1203.
13. Rossenblatt, F. (1958) The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review. 65:386-408.
14. Hagan, M. T.; Demuth,H. B.; & Beale, M. (1996) Neural Network Design. Boston: PWA Publishing Company.
15. Khairuddin, O.; Ramlan, M.; & Md. Nasir, S. (2000) Rangkaian Neural Genetik Aplikasi dalam Pengecaman Aksara Jawi. (Neural Network Genetic Application in recognizing Jawi characters) Dlm. Pertanika J. Sci. & Technol. 8(2):241-252,. Universiti Putra Malaysia Press
16. Fausett, L. Fundamental of Neural Networks: Architechtures, Algortihms, & Applications. New Jersey: Prentice Hall(1994).
17. Tsoi, A. C.; & Andersen, H. C. A Constructive Algorithm for the Training of a Multilayer Perceptron Based on the Genetic Algortihm. Department of Electrical Engineering, University of Quensland. (1994).
18. Trier, O. D., Jain, A.K. and Taxt, T (1996) Feature Extraction methods for character recognition- a survey. Journal of Pattern Recognition. Vol 29(4):641-662.