Jurnal Teknologi Maklumat & Multimedia 10(2011): 53 - 62

Performance Comparison of License Plate Recognition System Using Multi-Features and SVM

Siti Norul Huda Sheikh Abdullah
Marzuki Khalid
Khairuddin Omar

ABSTRACT

Feature extractor is one major factor in many image processing applications precisely in character recognition. The objective of this paper is to propose and to choose the best feature extractor for Malaysian licence plate recognition system. An enhanced Geometrical Feature Topological Analysis is proposed as a feature extractor and support vector machine is used as the classification technique. The proposed techniques and known feature extractors were used to justify its robustness for license plate recognition problem in precise. Previous research in the same domain, has applied straight pixels as the features. However, this approach is significantly acquire more time to execute the final recognition output typically in license plate recognition applications. Consequently, an alternative called the geometrical features with various combination techniques are proposed to enhance the overall performance in license plate recognition.

Keywords: License plate recognition, geometrical feature topological analysis, support vector machine

ABSTRAK

Pengekstrakan fitur merupakan satu faktor penting dalam aplikasi pemprosesan imej khususnya pengecaman aksara. Objektif kertas ini ialah untuk mencadangkan kaedah baru dan memilih kaedah pengesktrakan fitur terbaik untuk aplikasi pengecaman nombor plat kenderaan Malaysia. Satu penambahbaikan Analisis Topologi Fitur Geometri dicadangkan manakala mesin sokongan vektor digunakan sebagai kaedah pengkelasan. Kaedah cadangan dan beberapa kaedah pengekstrakan fitur dibandingkan untuk mengesahkan kelasakannya dalam masalah pengecaman nombor plat kenderaan secara khusus. Kebanyakan penyelidikan lepas menggunakan piksel asal sebagai fitur. Namun, kaedah tersebut meningkatkan masa larian keseluruhan pengecaman terakhir. Oleh itu, satu kaedah baru diperkenalkan dengan menggunakan fitur geometri dan gabungan teknik lain untuk meningkatkan prestasi keseluruhan pengecaman nombor plat kenderaan.

Kata kunci: Pengecaman nombor plat, Analisis Topologi Fitur Geometri, mesin sokongan keputusan

INTRODUCTION

Geometrical Feature Topological Analysis or simply known as GFTA is a statistical technique in text recognition. It combines the technique of zoning, thinning, geometrical features and/or contours. Some researches applied vertical, horizontal and diagonal and contour to extract important features after attempting the zoning approach based on GFTA. GFTA exclusively described four categories extracting and counting topological structures, measuring and approximating the geometrical properties, coding like chain coding Freeman approach, and graphs and trees such as strokes, loops, cross points which suitable for Arabic characters.

An extension of GFTA for applying curvature information such as number of circles, junction detections like X, Y, T junctions had also been introduced. Therefore in this research, additional information is adapted like maximum and minimum points of each zones and excluding ratio height to width information because each image has been resized similarly to find the best feature extractors and the best training model for classification in recognition module for license plate recognition (LPR) application. The experiments were conducted using license plate characters and numbers that had been collected in various states of Malaysia. Furthermore, GFTA has not been applied in any license plate application before.

This paper presents related works on LPR application using different approaches of feature extractors. This is followed by the proposed framework of LPR, the proposed description of GFTA theories, character classifications using Support Vector Machine, experiment configurations, results and analysis, and discussions.

RELATED WORKS

Remarkable color Dutch LPR study has been initiated and successfully accomplished 98.5% of recognition rate. This success was attained due to several deterministic approaches: Discrete-Time Cellular Neural Networks (DTCNN) in the feature extraction stage. An alternative method had slightly increased the performance up to 98.7% by introducing Hotelling transformation. Hotelling transform also known as Principal Component Analysis (PCA), Karhunen-Loeve or Eigen Vector Transformation. It is an advanced transformation technique which transforms one dimension and reconstructs the alphabet that has been found. However, other problem of PCA less prone to invariant therefore it always recognize number '8' as letter 'B,' 'I' as 'J' and 'O' as 'D' correspondingly. Meanwhile another common and simple method in feature extraction is called straight binarization or straight pixels. This approach which applied actual pixels as features had been widely used in license plate recognition applications [4, 5, 6]. Only straight pixel method had been examined in the experiments meanwhile PCA approach was not applied because our first attempt had shown that PCA was not achieving promising results. The following section will explain briefly the overall proposed framework.

THE CURRENT TECHNIQUES

GEOMETRICAL FEATURE TOPOLOGICAL ANALYSIS (GFTA)

The Geometrical Feature Topological Analysis (GFTA) was introduced by Janahiraman et al. (2002) and Tay et al. (1997), and also has been applied and modified by Cote et al. (1998), Nafiz et al. (2001) and Ning (2000). This technique actually employs information such as width and height of the bounding box, number of transitions between the object and background colors, values of gravity, the relative distance between the first and last points, the relative horizontal and vertical distances between first and last points, distance between two points, comparative lengths between two strokes, width of a stroke, upper and lower masses of words, and word length. Some researchers have used curvature information as its additional information such as number of circles, number of X points, number of Y points and number of T points. Table 1 shows the previous work referred to as GF 46 and GF 90 by Ning (2000) and Cote et al. (1998).

TABLE 1. GFTA of the previous techniques

	Number of horizontal or vertical zones	5	0	10
No	Type of Features	GF46	SP	GF90
1	Height, *H*	1	nil	1
2	Width, *W*	1	nil	1
3	Height to width ratio	1	nil	1
4	Center of gravity (X_{cg} and Y_{cg} positions)	2	nil	2
5	Total number of ink pixels, *I*	1	nil	1
6	Number of transitions, ζ	10	nil	20
7	Number of ink pixels, I_z	10	nil	20
8	Average *X* positions of ink pixel, *AveX*	10	nil	20
9	Average *Y* positions of ink pixel, *AveX*	10	nil	20
10	Min *X* and *Y* position of ink pixels, *MinX*, *MinY*	nil	nil	nil
11	Max and position of ink pixels, *MaxX*, *MaxY*	nil	nil	nil
12	Number of circle	nil	nil	1
13	Number of *X* point	nil	nil	1
14	Number of *T* point	nil	nil	1
15	Number of *Y* point	nil	nil	1
16	Original pixel values	nil	400	nil
	Total number of features, *fnum*	46	400	90

SUPPORT VECTOR MACHINE (SVM)

SVM is a well-known technique in machine learning precisely in character recognition and gene expression domains because it can easily separate linear or non-linear boundary.

The respective decision hyper surface in the 1-dimensional feature space is a hyperplane, that is where w = [W1, W2, …$_1$]^T is known as the weight vector and as the threshold or a constant in this high-dimensional space. Therefore the respective decision hyper surface in the 1-dimensional feature space is a hyperplane optimization that calculates approximations of the number of misclassified samples while is the tradeoff (penalty) parameter between error and margin and ξ is non-negative slack variable as Equation 1 until 4.

$$\min_{w} ij,\ _{b}ij,\ _{\xi}ij\ (w^{ij})^T\ w^{ij} + \mathrm{C}\ (\Sigma_t(\xi^{ij})_t). \tag{1}$$

$$\Big(\big((w^{ij})^T\ \phi(x_t)\big) + b^{ij}\Big) \geq 1 - \xi_t^{ij}. \tag{2}$$

$$\left(\left((w^{ij})^T \phi(x_t)\right) + b^{ij}\right) \geq -1 + \xi_t^{ij}. \tag{2}$$

$$\xi_t^{ij} \geq 0. \tag{4}$$

KERNEL FUNCTION

Radial Basis Functions (RBF) is chosen as the kernel function which acts as an inner product, or to really find a similarity measure between the objects. Function can take various forms like Equation 5.

$$K\left(x_i, x_j\right) = exp\left(-\gamma\|x_i, x_j\|^2\right). \tag{5}$$

THE PROPOSED TECHNIQUES

ENHANCED GEOMETRICAL FEATURE TOPOLOGICAL ANALYSIS (eGFTA) APPROACHES

In this research, additional information is adapted to find the best feature for characters classifications and may increase the accuracy rate. Before the image is segmented into five horizontal and vertical zones, the image of each characters were normalized into 20 × 20 pixel size. Later, each vertical zone and horizontal zones which have 4 × 20 and 20 × 4 matrix, respectively for each character are defined as in Figure 1.

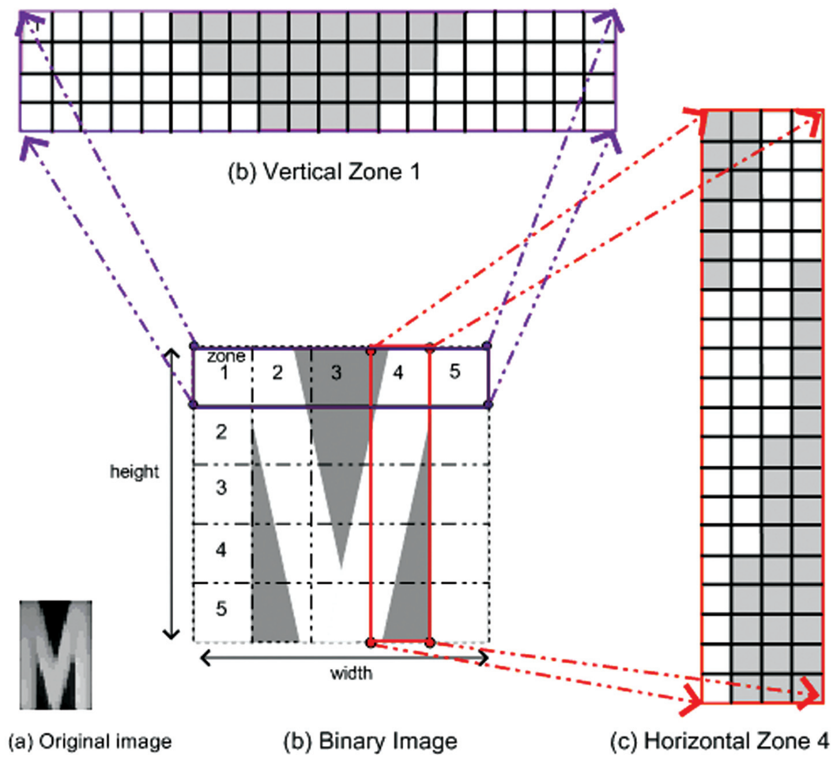


FIGURE 1. Example of (a) the original letter 'M' image which has been segmented (b) vertically and (c) horizontally

Table 2 shows the modification of the previous work referred to as *GF* 46 and *GF* 90 by. The total number of features, *fnum* for each feature type is also mentioned. This eGFTA has included (minimum and maximum *X* and *Y* points), excluded (height and width ratio, number of circles, 'X', 'Y' and 'T' points) and combined (straight pixels) some earlier features.

Explanation and example of each feature is calculated and described as follows:

1. Height: *H* means height of the character image. Example of *height* is 20 as shown in Figure 3.
2. Width: *W* means width of the character image. Example of *width* is 20 as shown in Figure 3.
3. Center of gravity: *Cg* consists of two features: *X* centroid (X_{cg}) and *Y* centroid (Y_{cg}), which are the center point of the foreground object (ink pixel or white pixel) in the character image.

X_{cg} and Y_{cg} are defined as Equation 6.

$$X_{cg} = \frac{\Sigma_{y=0}^{H-1} \Sigma_{x=0}^{W-1} f(x, y).\, x}{\Sigma_{y=0}^{H-1} \Sigma_{x=0}^{W-1} f(x, y)},$$

$$Y_{cg} = \frac{\Sigma_{y=0}^{H-1} \Sigma_{x=0}^{W-1} f(x, y).\, y}{\Sigma_{y=0}^{H-1} \Sigma_{x=0}^{W-1} f(x, y)}, \tag{6}$$

where $f(x, y)$ denotes the value of the pixel at position (x, y) as Equation 7.

$$f(x, y) = \begin{cases} 1 \; if\, f(x, y) = ink\_pixel, \\ 0 \; otherwise \end{cases} \tag{7}$$

Example of X_{cg} and Y_{cg} are 9.65 and 9.32 respectively are shown in Figure 3.

TABLE 2. The eGFTA techniques

No	Number of horizontal or vertical zones Type of Features	5 GF45	5 GF85	0 SP+5	5 SP+45	5 SP+85
1	Height, *H*	1	1	1	1	1
2	Width, *W*	1	1	1	1	1
3	Height to width ratio	nil	nil	nil	nil	nil
4	Center of gravity (X_{cg} and Y_{cg} positions)	2	2	2	2	2
5	Total number of ink pixels, *I*	1	1	1	1	1
6	Number of transitions, ζ	10	10	nil	10	10
7	Number of ink pixels, I_z	10	10	nil	10	10
8	Average *X* positions of ink pixel, *AveX*	10	10	nil	10	10
9	Average *Y* positions of ink pixel, *AveX*	10	10	nil	10	10
10	Min *X* and *Y* position of ink pixels, *MinX*, *MinY*	nil	20	nil	nil	20
11	Max and position of ink pixels, *MaxX*, *MaxY*	nil	20	nil	nil	20
12	Number of circle	nil	nil	nil	nil	nil
13	Number of *X* point	nil	nil	nil	nil	nil
14	Number of *T* point	nil	nil	nil	nil	nil
15	Number of *Y* point	nil	nil	nil	nil	nil
16	Original pixel values	nil	nil	400	400	400
	Total number of features, *fnum*	45	85	405	445	485

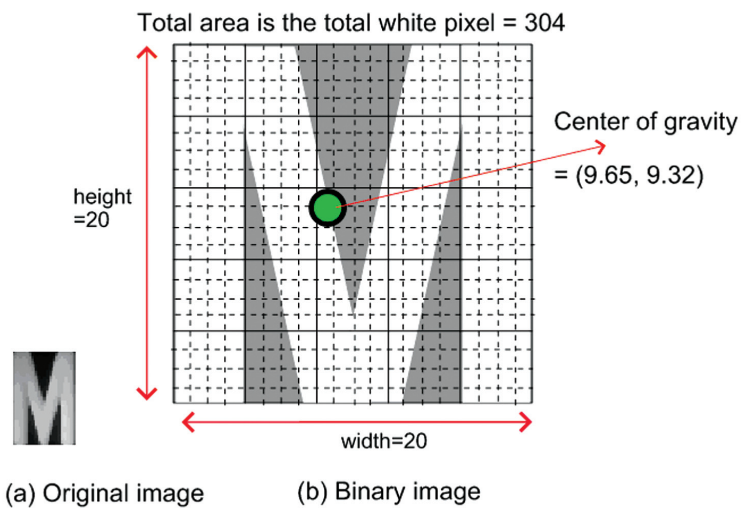


FIGURE 3. Samples of height, width, center of gravity and, and the total area of character 'M' image

4. Total number of ink pixels: *I* is total of ink pixels (foreground objects) of the character image. *I* is defined as Equation 8.

$$I = \Sigma_{y}^{H-1}\Sigma_{x}^{W-1} f(x, y) = 1. \tag{8}$$

Example of I is 304 as shown in Figure 3.

5. Number of Transitions: ζ is the number of transitions or changes from non-ink pixel to ink pixel in each vertical and horizontal zones, as shown in Figure 4 and 5, respectively. The number of transitions for the horizontal zone ζ_h is defined by:

$$\zeta^{h}_{i} = \tfrac{1}{n}\Sigma^{1}_{j=b}\tau_j , \tag{9}$$

and, the number of transitions for the vertical zone, is defined by:

$$\zeta^{v}_{i} = \tfrac{1}{n}\Sigma^{1}_{j=b}\tau_j , \tag{10}$$

Example of ζ_{v4} and ζ_{h1} are 1.25 and 2, respectively as shown Figure 4 and 5.

6. Number of ink pixels, I_z is the total ink pixels (foreground objects) of each horizontal, I_h and vertical zones, I_v. Example of I_{v4} and I_{h1} are 42 and 52 consecutively as shown Figure 4 and 5.

7. Average position of ink pixels for *X* and *Y*-axis: *AveX* and *AveY* are the average positions of ink pixels for *X* and *Y*-axis in each vertical and horizontal zones are shown as in Figure 4 and 5. The formula for *AveX* v_i and *AveY* v_i in vertical zone are defined in Equations 11 and 12 correspondingly.

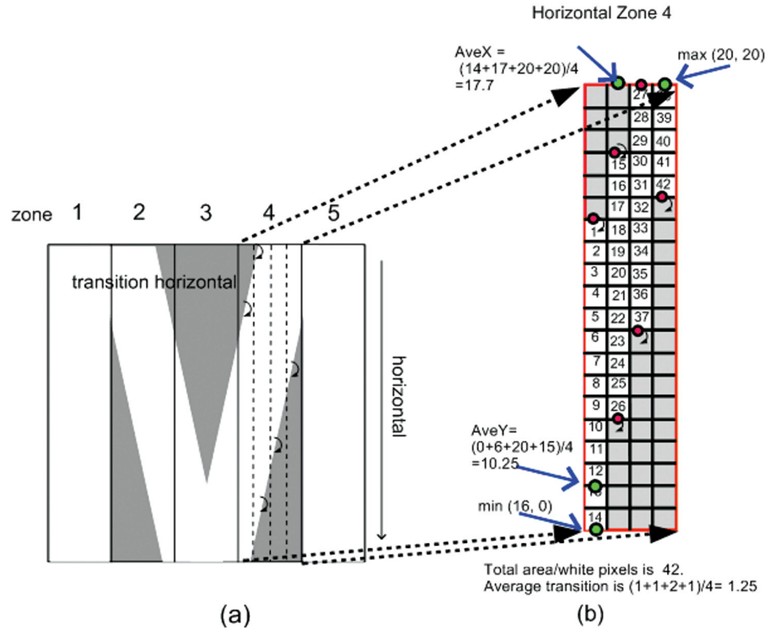


FIGURE 4. (a) The vertically segmented for character 'M' image, (b) the Vertical Zone 4 describes the transitions, number of pixels, average, maximum and minimum of X and Y values

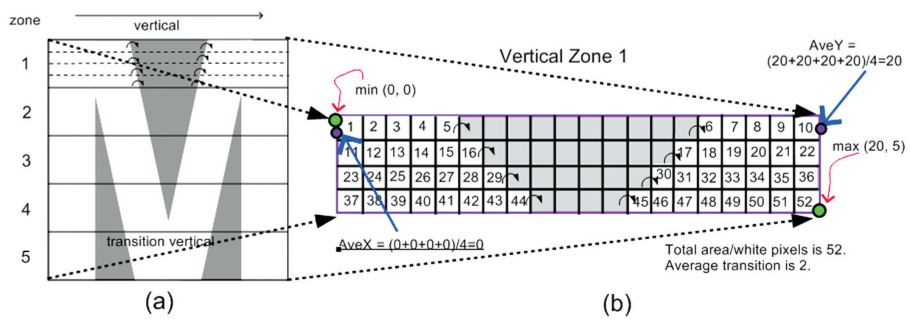


FIGURE 5. (a) The 'M' character image is horizontally segmented, (b) Horizontal Zone 1 describes the transitions, number of pixels, average, maximum and minimum for each X and Y values

$$AveX\,v_i = \tfrac{1}{n}\Sigma_{j=b}^{1} m_j . \quad (11)$$

$$AveY\,v_i = \tfrac{1}{n}\Sigma_{j=b}^{1} m_j . \quad (12)$$

Meanwhile the formula for and in horizontal zone are defined in Equations 13 and 14 correspondingly.

$$AveX\,h_i = \tfrac{1}{n}\Sigma_{j=b}^{1} m_j . \quad (13)$$

$$AveY\,h_i = \tfrac{1}{n}\Sigma_{j=b}^{1} m_j . \quad (14)$$

Example of $AveX_{v4}$ and $AveY_{v4}$ are 10.25 and 17.7, meanwhile, $AveX_{h1}$ and $AveY_{h1}$ are 1 and 20 consecutively as shown in Figure 4 and 5.

8. Minimum positions of ink pixels for X and axis Y: *MinX* and *MinY*, are the minimum position of ink pixels in each vertical and horizontal zones are shown as in Figure 4 and 5. The minimum position for the vertical zone $MinXv_i$ for X position is defined by:

$$MinXv_i = \begin{cases} X_s \; if \; X_s < X_{s+1}, \\ X_{s+1} \; otherwise, \end{cases} \quad (15)$$

Example of $MinX_{v4}$ and $MinY_{v4}$, are 16 and 1 consecutively meanwhile $MinX_{h1}$ and $MinY_{h1}$, are 1 and 1 correspondingly as shown Figure 4 and 5.

9. Maximum positions of ink pixels for X and axis Y: *MaxX* and *MaxY*, are the maximum positions of ink pixels in each vertical and horizontal zones, as shown in Figure 4 and 5, respectively. The $MaxXv_i$ for each vertical zones is defined as:

$$MaxX = \begin{cases} X_s \; if \; X_s > X_{s+1}, \\ X_{s+1} \; otherwise \end{cases} . \quad (16)$$

Example of $MaxX_{v4}$ and $MaxY_{v4}$ are 20 and 20 correspondingly meanwhile $MaxX_{h1}$ and $MaxY_{h1}$ are 20 and 5 consecutively as shown in Figure 4 and 5.

The minimum and maximum positions for the horizontal zones can be calculated using the same formula as the minimum and maximum positions for the vertical (Equations 15 and 17). The only difference is that the vertical zone requires the pixel to be scanned from top to bottom, instead of left to right for horizontal zone.

This is an example of the vertical eGFTA algorithm with 85 features. For the original features, every single pixel (total of 400) is inserted as features in an array. Figure 6 describes the whole structure of eGFTA. The eGFTA algorithm can be described as in Algorithm 1.

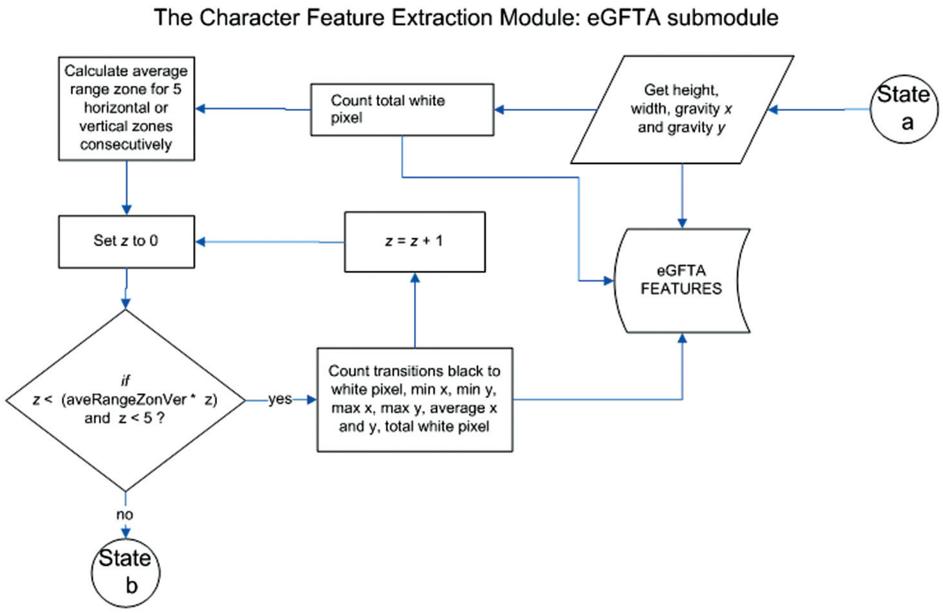


FIGURE 6. The flow chart of eGFTA submodule

ALGORITHM 1: The algorithm of eGFTA module for the LPR System

Input: Suppose the blob's height, H, and width, W, input image array after applying Median and Opening filters, $f(i, j)$ where, i and $j = 1, \ldots, H$ or W, gravity, (X_{cg}, Y_{cg}) and feature type, *featype*
Output: Obtain the feature array, $F(0, 1, \ldots, fnum - 1)$, where *fnum* is 85
1: $F_0 \Leftarrow H$, $F_1 \Leftarrow$ W. State a: Steps 1-32
2: Calculate the X_{cg}, Y_{cg} and total area, or the total of white pixels, I
3: Divide the image into five zones as vertically, v, and horizontally, h, according to the average range per zone, *RangeZonV*
4: start $\Leftarrow 0$, $end \Leftarrow 0$
5: While ($z < 6$) is true do
6: For ($start \leq i < end + 1$) is true do
7: For ($start \leq j < H + 1$) is true do
8: if $f(i, j) \neq f(i, j + 1)$ is true do
9: Count the transition, ζ_{v_z}, for every five zones
10: End if
11: if (($f(i, j)$ = white pixel or ink pixel) is true then
12: Obtain the number of ink pixels according for each zone, $I_{v_z}++$, the average maximum i and j point, $AveX_{v_{z,i}}$ and $AveX_{v_{z,j}}$, the minimum i and j point per zone, $MinX_{v_{z,i}}$ and $MinY_{v_{z,j}}$, the maximum i and j point per zone, $MaxX_{v_{z,i}}$ and $MaxY_{v_{z,i}}$
13: End if
14: Insert The I_{v_z}, $AveX_{v_{z,i}}$, $AveY_{v_{z,j}}$, $MinX_{v_{z,i}}$, $MaxX_{v_{z,i}}$, $MinY_{v_{z,j}}$ and $MaxY_{v_{z,j}}$ into $F_{5\ldots45}$
15: End for
16: End for
17: I_{v_z}, $AveX_{v_{z,i}}$, $AveY_{v_{z,j}}$, $MinX_{v_{z,i}}$, $MaxX_{v_{z,i}}$, $MinY_{v_{z,j}}$, $MaxY_{v_{z,j}} \Leftarrow 0$
18: $z++$
19: $end \Leftarrow end + RangeZonH$
20: End while
21: Divide the feature array into five horizontal zones, h
22: $start \Leftarrow 0$, $end \Leftarrow 0$
23: While ($z < 6$) is true do
24: For ($start \leq j < end + 1$) is true do
25: For ($start \leq j < W + 1$) is true do
26: Repeat steps 8 to 14
27: Insert I_{h_z}, $AveX_{h_{z,i}}$, $AveY_{h_{z,j}}$, $MinX_{h_{z,i}}$, $MaxX_{h_{z,i}}$, $MinY_{h_{z,j}}$ and $MaxY_{h_{z,j}}$ into $F_{46\ldots85}$
28: End for
29: End for
30: $z++$
31: $end \Leftarrow end + RangeZonH$
32: End while
33: Extract the next feature character. State b

RESULTS AND DISCUSSIONS

EXPERIMENT CONFIGURATIONS

The images used were Malaysian license plate (Figure characters and numbers (36 number of classes) which cover all roman letters (except O), numbers (from 0 to 9) and backslash (“/”). Examples of characters and numbers are illustrated as Figure 7.

(left) (right)

FIGURE 7. (left) Example of Middle East and Malaysian car images such as single and double line for standard, special, military and taxi. (right) Example of crop letters where (a) manually crop letters, (b) automatically cropped and (c) symbol, bulb and special license plates

The SVM scheme is as follows:

Input nodes	20 × 20 image pixel size	GF85, SP+GF5, SP+GF45, SP+ GF85, AKED
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

Initially, before training and testing 400 image sets, cross-validation was conducted to identify the best C and γ in order to obtain accurate training and testing data predictions. At the beginning, the chosen data used stratified sampling with three folds cross validation and the sample size increased proportionately as the numbers of feature. The overall results for best and γ pertaining to different feature selections are shown in Table 1.

There is one straight pixel values (*SP*), one previous technique (GF45), four proposed techniques (GF85, SP+GF5, SP+GF45, SP+GF85) and AKED. After obtaining the best C and γ as in Table 2, the data testing has been carried out on two different datasets, Dataset 1 and Dataset 2, in four parts: 1 – 2, and 2 – 1. For example, Testing 1 – 2defines as Dataset 1 is the training model and Dataset 2 is the training set.

TABLE 2. Selecting best C and γ values for support vector machine optimization

Feature type	C	γ	Accuracy
GF45	1024	0.3125	96.40%
GF85	1024	0.03125	93.80%
SP	32	0.009765625	97.60%
AKED (128+Kirsch)	32	0.000976563	95%

RESULTS AND ANALYSIS

This section is divided into two subsections which cover the overall performance and character analysis. The overall performance describes the correlations between datasets and various feature extractors. Meanwhile the character analysis section described the correlations between various feature extraction techniques and characters or numbers accuracy rates.

OVERALL PERFORMANCE

The results of the experiments are as in Figure 8 and Table 2. Another type of features extraction technique besides GF45, GF85, SP, SP+GF5, SP+GF45 and SP+GF85, AKED was also conducted. From this experiment, it can be concluded that GF45 is the best feature extraction technique with 99.49% while the second falls to GF85 with

99.36% accuracy rate. SP feature pixels is better than the AKED with 99.19% and 99.18% correspondingly.

Pertaining to Table 3, GF45 give better results compared to GF85. The differences of GF45 and GF85 accuracy rates in all the experiments are from 13 up to 43 percent. As a result, all data image sets has shown significant promising results in GF45 approach. Unlike GF45, some GF85 features such as the maximum or minimum points in each zone, are not significantly distinguishable.

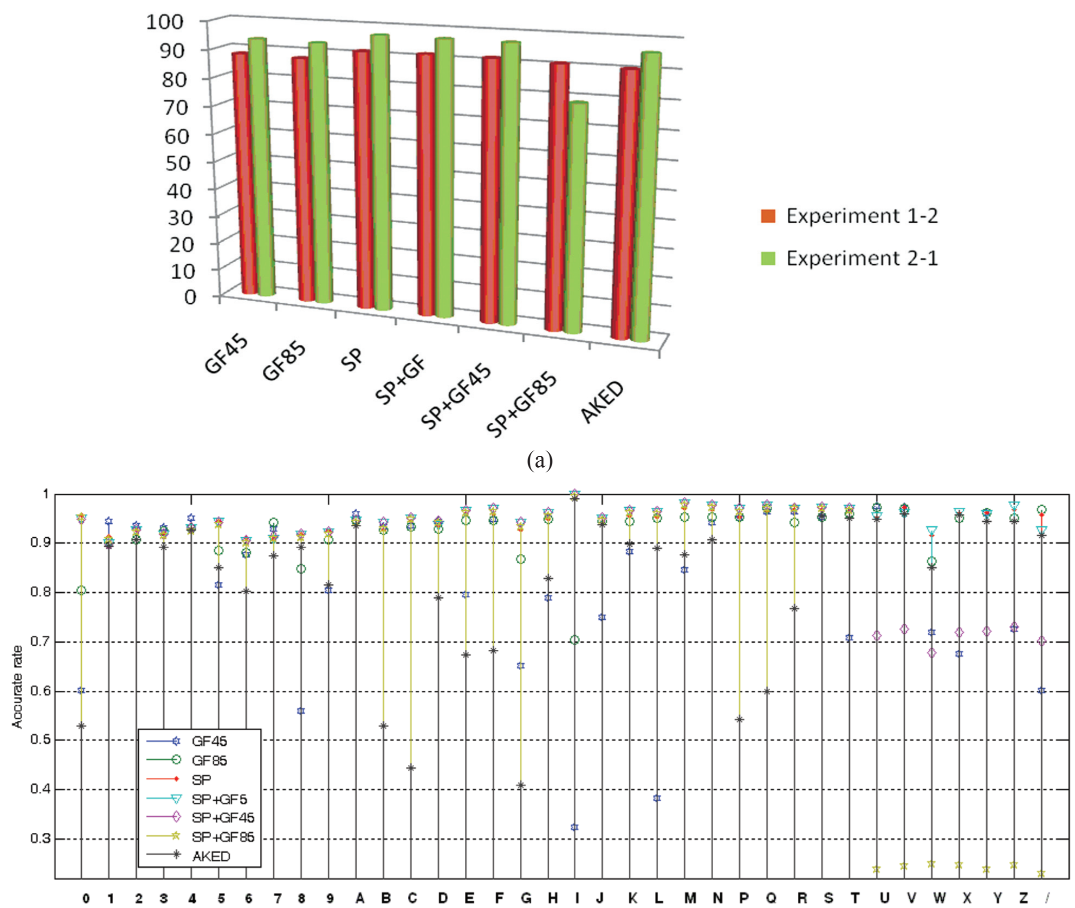
Despite that, SP approach or better known as straight pixels performs better than AKED. Even though, AKED might help to recognize corners or cursives or curve ends character much better than original pixel values but the results of this experiment is contradicted to. Details of character error analysis could give some justifications below.

TABLE 3. Experiment for eGFTA and AKED comparisons using SVM classification in percentage

Feature type	GF45	GF85	SP	SP+GF	SP+GF45	SP+GF85	AKED
1-2	88.33	87.9	91.49	91.71	91.69	91.04	90.58
2-1	93.69	93.44	97.18	97.08	97.08	78.63	96.06

Looking at the overall performance, Dataset 2 was the best SVM training model because almost all the recognition rate from Experiment 2-1, obtained more than 98%. The reason is the data image set 1 may contain more noisy image compared to image data set 2. Experiment 1-2, SP+GF5 approach gives the highest accuracy rate (91.71%) while SP, SP+GF45, SP+GF85 falls second (91.69%), third (91.49%) and fourth (91.04%) ranking correspondingly. Neither GF45 nor GF85 got higher than 90%. Consequently, this drop out may also due to less noisy and missing values in the image data 2 compared to image data set 1.

The results also suggested a new perspective on average accuracy rates for all techniques investigated in this SP approach gained the highest accuracy rate with 95.35% meanwhile SP+GF5, SP+GF45 and AKED obtained 95.07%, 95.05% and 94.83% correspondingly. This has proved that SPs contribute significant results because it maintains the actual pixel as features. It has a drawback of requiring extra processing time. Furthermore, AKED can also become a great competitor to either all of SP approaches because the accuracy rate only dropped slightly compared to solely eGFTA features.



(a)

(b)

FIGURE 8. Graphs of comparisons between eGFTA and AKED, (a) Accuracy rate comparisons, and (b) Character error analysis

CHARACTER ERROR ANALYSIS

Referring to Figure 8, each feature technique has its own strengths and weaknesses. Almost all techniques can identify all characters with fairly high accuracy rate; observed that SP and SP+GF5 achieved more than 90% while GF85 obtained more than 70%. However, SP+GF85 could hardly recognise characters like ‘V,’ ‘W,’ ‘X,’ ‘Y,’ ‘Z’ and ‘/.’ Meanwhile AKED had significantly failed to detect curving characters like ‘0,’ ‘B,’ ‘C,’ ‘G,’ ‘Q’ and ‘R.’ Unfortunately, this result does not agree with earlier findings where AKED using MLP-BP could recognize cursive characters better than other edge detector kernels.

Even though GF45 was the best technique, it has weaknesses recognizing characters ‘I’ and ‘L’ because its accuracy rates were 0.33 and 0.39, correspondingly. Combination of SP+GF45 has shown better performance than GF45 but slightly dropped in detecting ‘V,’ ‘W,’ ‘X,’ ‘Y,’ ‘Z’ and ‘/’ letters with accuracy rate in the range of 0.68 to 0.75. As a conclusion, despite that GF45 and GF85 obtained better results in Figure 8(a) and Table 2 but they does not mean they could recognize all characters consistently. SP and SP+GF5 has shown acceptable performance but unfortunately could not increase the overall processing time due to increasing number of features. An example of the LPR system using SP technique is given in Figure 9 where the letter ‘B’ was commonly guessed as number ‘8.’ It is suggested that other techniques such as Bag of Features, Fourier Transform or Trace Transform be incorporated to boost up the performance of character recognition module.

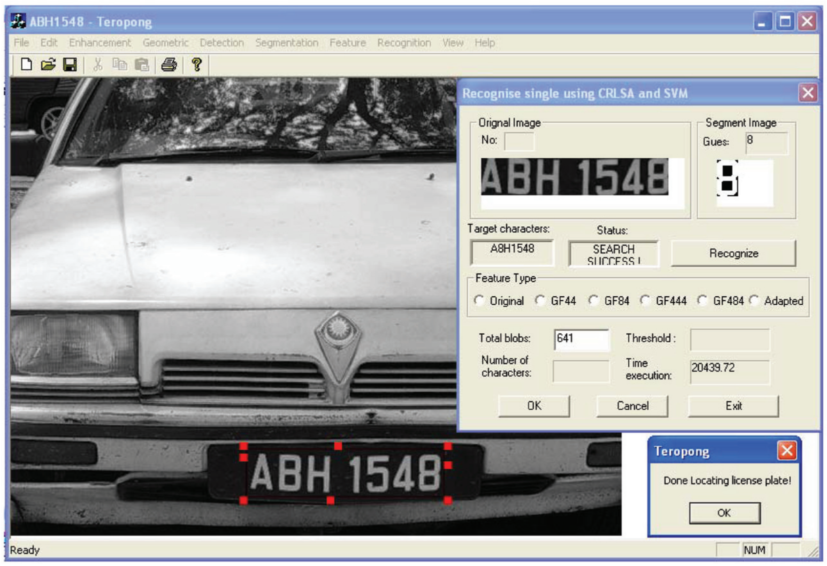


FIGURE 9. A sample of recognition error where letter ‘B’ is guessed as number ‘8’ using SP features and SVM

CONCLUSIONS

The advantages of AKED and eGFTA can be described as follows:

1. AKED can distinguish cursive characters better than eGFTA.
2. eGFTA requires less memory allocation because the number of features is less than AKED.
3. The processing time of eGFTA is lesser than AKED.
4. The eGFTA approach does not change the actual pixels of the image. Therefore the image pixels are consistent compared to AKED.
5. The AKED are more robust in terms of solving ‘salt and pepper’ image problem.

The disadvantages of AKED and eGFTA would be as follows:

1. AKED require much memory allocation.
2. AKED and eGFTA approach less sensitive to invariant characters. Therefore, the license plate’s symmetry is advised to handle first.
3. The computation of eGFTA approach is more complex than AKED.
4. The processing time of AKED are longer than eGFTA.
5. The eGFTA approach may not robust for ‘salt and pepper’ image problem.

As conclusion, this paper has presented a new development in LPR based on variety of feature extraction techniques and the SVM pattern classifier which has not been well used yet in many applications. The research has shown that GF45 feature extractor was rather suitable for such application with high accuracy and advantage in recognizing cursive characters. Meanwhile GF85 and SP+GF45 approaches, have its own advantage because they can almost gained similar accuracy rates as SP. However, if processing time is not the critical issue, than SPs and AKED approaches can also give a competitive results in the overall performance of LPR application. Besides that, s actually achieved more consistent in terms of character error analysis. Another point to highlight is the data model selection. Image dataset 2 is considered the best SVM training model.

ACKNOWLEDGEMENTS

We wish to thank Pattern Recognition research group and David Al Dabass for their support and cooperation in developing the system. We would also like to thank the University for the research grant UKM-GGPM-ICT-119-2010 entitled "Text and logo recognition for vision guided" and UKM-TT-03-FRG0129-2010 entitled "Determining adaptive threshold value for image segmentation."

REFERENCES

Chih-Wei Hsu & Chih-Jen Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks* 13.

Cote, M., Lecolinet, E., Cheriet, M. & Suen, C.Y. 1998. Automatic reading of cursive scripts using a reading model and perceptual concepts. *International Journal of Document Analysis and Recognition*, 1(1): 3-17.

Demetrios Michalopoulos & Chih-Kang Hu. 2002. An Error Back-Propagation Artificial Neural Networks Application in Automatic Car License Plate Recognition, volume 2358 of Lecture Notes in Computer Science, chapter Developments in Applied Artificial Intelligence: 15th International Conference on Industrial and Engineering. Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2002, Cairns, Australia, June 17-20, 2002. Springer-Verlag Berlin Heidelberg.

Janahiraman, T., Tay, Y.H., Khalid, M. & Yusof, R. 2002. Classification of Handwritten Digits Using Geometrical and Topological Feature Extraction Technique and Neural Network. Proc of Int'l Conf on Artificial Intelligence in Engineering & Technology (ICAIET2002), Kota Kinabalu.

John Fulcher. 2006. *Advances in Applied Artificial Intelligence*. United States of America & United Kingdom: Idea Group Publishing 2006.

Nafiz Arica & Fatos T. Yarman-Vural. 2001. An Overview of Character Recognition Focused on Off-Line Handwriting. IEEE Transactions On Systems, Man, And Cybernetics–Part C: Applications And Reviews 31(2): 216-234.

Nijhuis, J.A.G., Brugge, M.H. & Helmholt, K.A. 1997/1998. License Plate Recognition Using DTCNNs. Proceedings 1998 Fifth IEEE International Workshop on Publish Security Technology, 212-217.

Ning Lim Woan. 2000. Design an automated data entry system hand-filled form. PhD thesis, Universiti Teknologi Malaysia.

Siti Norul Huda Sheikh Abdullah, Marzuki Khalid, Rubiyah Yusof & Khairudin Omar. 2007. Comparison of feature extractors in license plate recognition. IEEE Proceedings of First Asia International Conference on Modelling & Simulation (AMS2007), Phuket, Thailand.

Taikyeong Jeong, Jinsuk Kang & Yeon Sung Choi. 2006. Implementation of Embedded System for Intelligent Image Recognition and Processing, volume 3980/2006 of Lecture Notes in Computer Science, chapter Computational Science and Its Applications - ICCSA: 993-999. Springer-Verlag Berlin Heidelberg.

Tay, Y.H. & Khalid, M. & Tan, K.K. & Yusof, R. 1997. Hand-written Postcode Recognition by Fuzzy ARTMAP Neural Network. Proceedings of Confederation Of Scientific and Technological Associations in Malaysia – COSTAM's Malaysian Science and Technology Congress '97.

Theodoridis, S. & Koutroumbas, K. 2003. Pattern Recognition of 012685875690090. United States of America: Elsevier Academic Press.

Trier, O.D., Jain, A.K. & Taxt, T. 1996. Feature Extraction methods for character recognition-a survey. *Journal of Pattern Recognition*, 29(4): 641-662.

Wintz, Paul A. 1972. Transform Picture Coding. *Proceedings of the IEEE*, 60(7): 809.

Yao-Quan Yang, Jie Bai Rui-Li Tian & Na Liu. 2006. Research of Vehicle License Plate Location Algorithm Based on Color Features and Plate Processions, volume 3930 of Lecture Notes in Computer Sciences, chapter Advances in Machine Learning and Cybernetics. Springer-Verlag Berlin Heidelberg.

Siti Norul Huda Sheikh Abdullah
Khairuddin Omar
Center for Artificial Intelligence and Technology (CAIT),
Faculty of Information System and Technology,
Universiti Kebangsaan Malaysia,
43600 UKM, Bangi, Selangor
mimi@ftsm.ukm.my,
http://www.ftsm.ukm.my/cait

Marzuki Khalid
Centre for Artificial Intelligence and Robotics (CAIRO),
Faculty of Electrical Engineering,
Universiti Teknologi Malaysia,
Jalan Semarak, 54100 Kuala Lumpur
marzuki@utmkl.utm.my
http://cairo.utmkl.utm.my